RSX–11M–PLUS/RMS–11

Release Notes

Order No. AA-H427C-TC

RSX–11M–PLUS Version 2.1
RMS–11 Version 2.0

digital equipment corporation · maynard, massachusetts

First Printing, October 1979
Revised, April 1982
Revised, April 1983

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by Digital Equipment Corporation or its affiliated companies.

Printed in U.S.A.

The postpaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist in preparing future documentation.

The following are trademarks of Digital Equipment Corporation:

DEC	DIBOL	RSX
DEC/CMS	EduSystem	UNIBUS
DEC/MMS	IAS	VAX
DECnet	MASSBUS	VMS
DECsystem-10	PDP	VT
DECSYSTEM-20	PDT	digital
DECUS	RSTS	
DECwriter		

ZK2256

HOW TO ORDER ADDITIONAL DOCUMENTATION

In Continental USA and Puerto Rico call 800-258-1710

In New Hampshire, Alaska, and Hawaii call 603-884-6660

In Canada call 613-234-7726 (Ottawa-Hull)
800-267-6146 (all other Canadian)

DIRECT MAIL ORDERS (USA & PUERTO RICO)*

Digital Equipment Corporation
P.O. Box CS2008
Nashua, New Hampshire 03061

*Any prepaid order from Puerto Rico must be placed with the local Digital subsidiary (809-754-7575)

DIRECT MAIL ORDERS (CANADA)

Digital Equipment of Canada Ltd.
940 Belfast Road
Ottawa, Ontario K1G 4C2
Attn: A&SG Business Manager

DIRECT MAIL ORDERS (INTERNATIONAL)

Digital Equipment Corporation
A&SG Business Manager
c/o Digital's local subsidiary or approved distributor

Internal orders should be placed through the Software Distribution Center (SDC), Digital Equipment Corporation, Northboro, Massachusetts 01532

CONTENTS

CONTENTS

PREFACE

MANUAL OBJECTIVES

This manual summarizes the new software features for both RSX-11M-PLUS V2.1 and RMS-11 V2.0. It also includes information on problems, restrictions applicable to the new software, and incompatibilities with the previous version. Therefore, it is important to read this document before generating and using your system.

INTENDED AUDIENCE

The RSX-11M-PLUS/RMS-11 Release Notes are intended for all users of the RSX-11M-PLUS operating system.

STRUCTURE OF THIS DOCUMENT

CHAPTER 1 RSX-11M-PLUS V2.1 RELEASE NOTES

Section 1.1 New Features for RSX-11M-PLUS V2.1

Describes briefly the major new features of RSX-11M-PLUS V2.1.

Section 1.2 Summary of Technical Changes

Provides brief summaries of the new software features and refers you to the appropriate manuals in the documentation set for complete information. Any changes made to the documentation since V2.0, including new, revised, or updated manuals, are also indicated.

Section 1.3 Notes on Software

Describes the incompatibilities between the previous and the new versions of RSX-11M-PLUS, and provides additional miscellaneous information pertinent to the successful use of the software.

Section 1.4 Reporting Problems and Applying Corrections

Provides information on reporting any problems you encounter with the new software and describes procedures for applying corrections to files on the source disk.

CHAPTER 1

RSX–11M–PLUS VERSION 2.1 RELEASE NOTES

CHAPTER 1

RSX-11M-PLUS VERSION 2.1 RELEASE NOTES

1.1 NEW FEATURES FOR RSX-11M-PLUS V2.1

RSX-11M-PLUS V2.1 software is an update of the disk-based RSX-11M-PLUS operating system. The major new features for RSX-11M-PLUS V2.1 include the following:

- Software performance monitor support
- Support for Micro/PDP-11
- New device support
- Multivolume backups for disks
- Increased pool size
- RC25 pregenerated disk distribution kit[1]
- Support for overlays in tasks with separated instruction and data space (I- and D-space)
- RMS-11 Version 2

1.1.1 SYSGEN Features

The following SYSGEN features are new for V2.1:

- A new question asks whether the user wants software performance monitor (SPM-11) support. SPM-11, which must be purchased separately, provides data on system usage. This question is always asked, even if the user chooses the Full-functionality Executive. If the user answers Yes, the SPM-11 hook points are included in the system. You can then install the separate SMP-11 layered product.
- The Micro/PDP-11 has been added to the list of supported processors.
- If you include a DU device (RA60, RA80, RA81, RC25, RD51, and RX50) in your system, SYSGEN automatically builds, installs, and fixes VER, the remount verification task (see Section 1.1.8, RC25 Disk Subsystem).

1. There are a number of references in the RSX-11M/M-PLUS manuals to the RC25 device. This device, as of the first publication date of the manuals, is not available for sale or distribution; nor is its mention in the documents a commitment by DIGITAL to sell or distribute this device.

- Devices with the mnemonic DU are now valid crash devices. As with all devices, crash support is restricted to removable-media devices.

- SYSGEN now supports the following new devices:

DHV11	RA81
LA50	RC25
LN01	RD51
LP07	RX50
LP27	TSV05
RA60	TU80

1.1.2 Additions to MCR

MCR supports new disk devices and has new or expanded keywords for the OPEN and SET commands. See Section 1.2.2 for a complete list of new MCR features.

1.1.3 Backup and Restore Utility (BRU)

BRU has three new features for RSX-11M-PLUS V2.1: multivolume backups for disks, a new sort algorithm, and additional device support. See Sections 1.2.4 and 1.3.19 for additional information.

1.1.4 Task Builder (TKB)

The following is an abbreviated description of the new functionality of the Task Builder for RSX-11M-PLUS V2.1.

- The Autoload entry point has been modified. This entry point is now accessed by an indirect reference through the overlay run-time system's impure area, rather than by direct reference in the Autoload vector.

- For user mode instruction and data space (I- and D-space) tasks, the Autoload vectors consist of two parts:

  - An I-space part consisting of 4 words contained in the program section, $$ALVI.

  - A D-space part consisting of 2 words contained in the program section, $$ALVD.

- The symbol table (.STB) file now contains two forms of symbol definition. To maintain backward compatibility, all autoloadable symbols are entries in the global symbol directory, and the vector itself is defined in associated text records. Additionally, a new internal symbol directory record, which is TKB-generated, supports selective inclusion of autoloadable symbols for conventional tasks as well as the information needed to generate autoload vectors for I- and D-space tasks.

- Autoload vectors supplied by the symbol table (.STB) files of the resident libraries are now selectively included in the user's task image. Previously, all autoload vectors resulting from autoloadable symbols were present in the task image. Now, only referenced symbols result in autoload vectors that are present in the task image.

- The size of the segment descriptor block that is internal to TKB depends on whether an I- and D-space task is being built. For a conventional task, the size of the internal segment descriptor is 230(8) bytes. For an I- and D-space task, the size is 262(8) bytes.

- The task-resident segment descriptors for a user mode I- and D-space task are 32(8) bytes in length. The first 22(8) bytes of the segment descriptor are identical for conventional and I- and D-space tasks. However, the low-order bit of the second word is 0 for conventional tasks and 1 for I- and D-space tasks. For D-space segment descriptors, the last 10(8) bytes contain the disk block address, the virtual load address, the segment length in bytes, and the window descriptor address. The segment length is 0 if D-space is not present in the segment.

- Disk space allocation for I- and D-space overlay segments is large enough to accommodate the I-space part and the D-space part in separate but adjacent disk blocks. The I-space part comes first on disk, and the D-space part comes second. When an overlay segment is read into memory, the I- and D-space parts are read separately. However, the speed with which I- and D-space parts are read into memory is increased because these parts are adjacent on the disk.

- The rules concerning cluster libraries have changed so that the first library of the cluster, the default library, can now have a non-null root.

- The following option is new in TKB:

  DSPPAT Allows object-level patching of a conventional task or the D-space part of an I- and D-space task.

- The following TKB switch is new:

  /EL The Extend Library switch specifies the maximum possible size for the library according to the size specified in the PAR option. The switch specifies a larger library virtual address range than is actually present in the library, in order to allow RMS to map its vectored library segments.

1.1.5 RSX-11M-PLUS Executive

Pool space has been increased by moving several Executive modules into the directive commons region. Taking these modules out of the Executive adds approximately 2200(10) bytes to the dynamic storage region of a system with a standard Executive containing only Instruction space. The affected modules and their respective entry points are as follows. Note that ERSUB is a new module that was originally part of ERROR.MAC.

Modules	Entry Points
DRCMT	$DRCMS $DRCMT
DRDCP	$DRDCP $DRECP

Modules	Entry Points
DRRES	$DRUNS
	$DRRES
	$DRSTP
	$DRSPN
	$DRATP
DRSED	$DRREF
	$DRCEF
	$DRRAF
	$DRSEF
	$DRSTS
	$DRSTL
	$DRWFL
ERSUB	$DTOER
	$DVTMO
	$DVERR
	$DVCER
	$LOGER
	$FNERL
	$CRPKT
	$QUPKT
	$QERMV

1.1.6 New Device Support

A number of new devices are supported for RSX-11M-PLUS V2.1. See Section 1.2.5 for a complete list.

1.1.7 Pregenerated Kits

Prior to RSX-11M-PLUS V2.1, the RL02 disk distribution kit was the only pregenerated kit. For V2.1, another pregenerated disk distribution kit has been added, the RC25.

The RL02 disk kit for V2.1 differs from that of previous versions in three ways. Note that the following three features also apply to the RC25.

First, the RL02 and RC25 disk kits contain two systems. The first system, found in UIC [2,54], supports I- and D-space and supervisor mode. This I- and D-space system is intended for use on PDP-11/44 and PDP-11/70 processors. The second system, found in UIC [1,54], does not support I- and D-space and supervisor mode. This non-I- and D-space system is intended to be used on Micro/PDP-11, PDP-11/23-PLUS, and PDP-11/24 processors. As distributed, the non-I- and D-space system is the hardware-bootable system.

Note that you can reclaim disk space by deleting the unused system. The indirect command file [1,2]DELETESYS.CMD, which deletes the unused system, is supplied on the pregenerated kits. See Chapter 5 of the RSX-11M-PLUS System Generation and Installation Guide for specific instructions.

Second, to increase pool space for the non-I- and D-space system, certain tasks have been removed from this system. This system does not contain shadow recording, console logging, memory parity, and Executive support for the SPM-11 performance monitor.

Although parity memory support was removed from the non-I- and D-space system, this system will run on systems with parity memory. However, the non-I- and D-space system does not log parity errors or isolate failed memory.

Third, there is a new, easily edited STARTUP.CMD file.

For a full explanation of pregenerated distribution kits, see Chapter 5, Pregenerated RSX-11M-PLUS Kits, in the RSX-11M-PLUS System Generation and Installation Guide. Also see Section 1.2.1 of these Release Notes.

1.1.8 RC25 Disk Subsystem

The RC25 is a new disk subsystem supported on RSX-11M-PLUS V2.1. The following sections describe the RC25.

1.1.8.1 **RC25 Hardware Description** - The RC25 disk subsystem consists of a fixed-media drive and a removable-media drive, both of which revolve on the same spindle and share the same head mechanics. Each drive is a logical unit, so each RC25 disk subsystem consists of two logical units. The UDA50 hardware controller (DU) can support two subsystems or four logical units.

1.1.8.2 **Stalling Input and Output (I/O)** - Since two RC25 disk units revolve on the same spindle and share the same head mechanics, you must spin down both units of a subsystem together in order to spin down one unit. You cannot access either unit until both are spun up again. To compensate for spinning down both units when you only want to spin down one, the device driver (DUDRV) postpones input and output to both units until the device is spun up again and the heads are reloaded. This is called stalled I/O.

Stalling I/O to an RC25 subsystem affects the system's performance. If you initiate an operation requiring I/O to a stalled unit, you will not receive a timely response to the request. Although the I/O request is queued to the device driver, the driver ignores the request until the drive is loaded and the remount verification task verifies that the unit is ready (see Section 1.1.8.4). The driver then resumes processing requests. Note, however, that an operation can continue as long as it does not require access to the unit whose I/O is stalled.

Sometimes an operation that does not involve stalled-I/O units is delayed as well. For example, assume that your system disk is in the fixed-media unit and that you spin down a subsystem in order to change the disk pack in the removable-media unit. If a user then initiates an operation requiring a task to be loaded from the fixed unit, the loader issues a queued I/O request to the fixed unit. However, the device driver does not respond to this request immediately, since the subsystem is spun down. Also, because the loader cannot service additional tasks until it loads the current task from the disk, load operations to other disks on the system remain in the loader's work queue until the current load operation completes.

NOTE

Like the loader, the Files-11 Ancillary Control Processor (Files-11 ACP or F11ACP) is another single-threaded task that may delay response time when I/O is stalled to the RC25. To avoid this delay, you should always install a unique ACP for the RC25 units (see the MOU command in the RSX-11M/M-PLUS MCR Operations Manual).

System users may find it difficult to distinguish between system crashes and system delays due to stalled I/O. Therefore, it is recommended that, before you spin down an RC25 subsystem, you inform all system users of your intentions.

Note that I/O is not stalled for volumes mounted foreign. In order not to lose I/O requests to a foreign volume, you should dismount the volume before you spin down an RC25 subsystem. For example, if you intend to remove the disk from the removable-media unit, you should dismount the foreign volume in the fixed-media unit before you spin down the device. (See Section 1.1.8.3, Dismounting the RC25.) If you spin down the device without dismounting the foreign volume, the driver rejects additional I/O requests to that volume and issues the error message "Device not ready."

If you receive this error message, dismount the foreign volume, remount the volume as foreign, and start the operation again.

1.1.8.3 **Dismounting the RC25** - You dismount a unit on the RC25 in the same way as for other disk devices, by using the DISMOUNT command. As a nonprivileged user, you can issue the DISMOUNT command to logically disconnect an RC25 unit. However, you must be privileged in order to spin down the device while dismounting it, that is, to use the /UNLOAD qualifier for the DISMOUNT command. The privileged status of DISMOUNT/UNLOAD is a safety measure to control who is able to spin down the system disk.

If you are a privileged user, DISMOUNT/UNLOAD issues the following message when the command executes properly:

```
Warning -- All units of multiunit drive will spin down <ddnn:>
```

If you are a nonprivileged user, DISMOUNT/UNLOAD refuses your request to spin down a unit and issues the following message:

```
Warning -- Volume will not spin down <ddnn:>
```

1.1.8.4 **Remount Verification Task** - After you spin up the units of an RC25 subsystem, the remount verification task (task name VER...) checks the unit for which I/O was stalled. VER... verifies that the same volume is still mounted on this unit by checking that the volume label, the serial ID, and the home block checksums of the memory resident data structure (the Volume Control Block) match the on-disk structure.

VER... issues the following message to the console terminal each time it begins to check a unit:

Starting verification on unit DUnn:

If the verification is successful, VER... issues the following message to the console terminal:

Verification complete on unit DUnn:

Once the remounting process completes, the driver resumes processing I/O requests to that unit.

If the verification fails, VER... issues the following message:

Verification failed on unit DUnn:

If the verification fails on a unit, I/O continues to stall. The driver stalls I/O requests until VER... determines that the unit contains the original volume.

If you are using the fixed-media unit of an RC25 as your system disk, you need to fix the remount verification task in memory. The remount verification task is built during SYSGEN when you select any Mass Storage Control Protocol (MSCP) device. SYSGEN creates a skeleton VMR command file that includes commands to install and fix VER... in your system.

After you spin down and then spin up an RC25 subsystem, the driver checks to see if the remount verification task is installed. If the driver does not find VER... installed in the system, it notifies TKTN to issue the following message:

<ddnn:> -- Remount verification task not installed

If VER... is not installed and you spin down an RC25 subsystem, you can correct this error as follows.

- If the RC25 unit that you spin down contains your system disk:

  1. Reboot the system.

  2. Install VER.TSK to prevent this problem from occurring again.

- If the RC25 unit that you spin down does not contain your system disk:

  1. Install VER.TSK.

  2. Spin the unit down again, and then spin it up, in order to notify the driver that VER... is now installed.

1.1.9 Files-11 Ancillary Control Processor (Files-11 ACP)

Two new features have been added to Files-11 ACP (F11ACP) in RSX-11M-PLUS V2.1 to support RMS-11 V2.0.

- F11ACP supports new file attribute codes (creation, revision, and backup dates), which read and write the date in the file header using a 64-bit date format. F11ACP performs a computation to convert the date between ASCII and 64-bit binary form.

- F11ACP also supports the $SEARCH function of RMS-11 V2.0 by providing character wildcarding assistance. This support substantially reduces processing time when portions of a filename include wildcards.

Note that because of these features, you cannot use RMS-11 V2.0 on a version of RSX-11M-PLUS prior to V2.1.

1.1.10 Support for the LA50, LA100, and LN01 Printers

Support has been added to RSX-11M/M-PLUS for the LA50, LA100, and LN01 printers. These printers are capable of printing either 132 or 80 columns on 8 1/2 inch wide paper.

You can select support for these printers while performing SYSGEN. During SYSGEN, this support is a task-build option for the queue manager (QMG) and the line printer processor (LPP), and it is the default.

In order to use an LA50, LA100, or LN01 printer, you set the printer type when you initialize the print processor by using a form type in the command line. The valid form types for initializing the print processor are as follows:

Form types	Printer type
125.	LA50
126.	LA100
127.	LN01

For example, if you have a LA50 printer connected to terminal line TT6:, you use the following command to initialize the print processor.

```
(MCR)     QUE TT6:/SP/FO:125./FL:1/LOWER

(DCL)     INITIALIZE/PROCESSOR TT6:/FORMS:125/FLAG:1/LOWER
```

When you wish to print a file, select your desired printer mode by including one of the following form types in the command line:

Form types	Printer mode
3.	132-column, draft quality. Also landscape mode for LN01.
4.	80-column, letter quality. Also portrait mode for LN01.
5.	132-column, letter quality. Also landscape mode for LN01.
6.	80-column, draft quality. Also portrait mode for LN01.

For example, to print the 132-column draft quality TEST.LST file (that is, a normal listing file) and the 80-column letter quality TEST.TXT file on an LA50, use the following commands. Note that these commands assume that you have assigned the default print queue, PRINT, to the processor handling the LA50.

```
(MCR)     PRI /FO:3.=TEST.LST
          PRI /FO:4.=TEST.TXT

(DCL)     PRINT/FORM:3 TEST.LST
          PRINT/FORM:4 TEST.TXT
```

Modifying or Removing Support for the Printers

The printer options are controlled by task-build paramenters in the build files for the queue manager and the line printer processor. These build files are, respectively, [1,20]QMGBLD.BLD and [1,20]LPPBLD.BLD on the distribution kits. Prior to building the queue manager and the line printer processor during SYSGEN, you can modify these files for the following reasons:

1. You can change the form types that are used to represent printer types. To do so, modify the GBLDEF options that define the symbols Q$LSPS and Q$HSPS. The comments in the build files explain the values of the symbols. The normal defaults are 125. through 127.

2. You can change the form types that are used to represent document types. To do so, modify the GBLDEF options that define the symbols Q$LSPF and Q$HSPF. The comments in the build files explain the value of the symbols. The normal defaults are 3. through 6. If you modify the range of form types, you must change the GBLPAT options that define the forms to include offsets appropriate for the new form types. (See the comments in the command file regarding form definition).

3. You can remove the support for these printers altogether. This may be necessary if you do not have any of these printers and if you use form types in the range 3. through 6. or 125. through 127. In order to remove the support, set the values of Q$LSPS and Q$LSPF to 1 and the values of Q$HSPF and Q$HSPS to 0. You may also wish to remove the four GBLPAT definitions for the form types 3. through 6.

Note to LA100 Users

If you have a LA100 printer and you use it for standard 132-column, 14 1/2 inch paper, you probably will wish to establish it as your standard printer --with form type 0--and not use this support. This support is intended primarily for applications including all three printers and involving 8 1/2 inch wide forms.

1.1.11 PDP-11 PASCAL/RSX

RSX-11M-PLUS V2.1 includes DCL support for PDP-11 PASCAL/RSX. Note, however, that PDP-11 PASCAL/RSX is scheduled to be released at a date later than RSX-11M-PLUS V2.1. Therefore, until PDP-11 PASCAL/RSX is available on your system, the use of the PASCAL command will result in the informational message "PASCAL -- Sorry, task not installed."

RSX-11M-PLUS V2.1 lists PASCAL as one of the available HELP topics. However, until PDP-11 PASCAL/RSX is released and becomes available on your system, typing "HELP PASCAL" will result in the message "PASCAL currently is unavailable on your system."

PASCAL commands will be documented in the PASCAL User's Guide, distributed at the time of the PDP-11 PASCAL/RSX release. Complete HELP files for PDP-11 PASCAL/RSX also will be provided with the PASCAL distribution kit at the time of its release.

1.2 SUMMARY OF TECHNICAL CHANGES

The following sections summarize the new software features for RSX-11M-PLUS. A summary of technical changes also appears in the preface of each manual.

The title of each manual is followed by a letter indicating whether the manual is a new (N), revised (R), or updated (U) version.

1.2.1 RSX-11M/M-PLUS System Generation and Installation Guide (R)

The revision of the RSX-11M-PLUS System Generation and Installation Guide reflects the following software technical changes and additions:

- Support has been added for the following devices:

  DHV11
  LA50
  LN01
  LP07
  LP27
  RA60
  RA81
  RC25
  RX50
  RD51
  TU80
  TSV05

- The Micro/PDP-11 has been added as a supported processor.

- Support has been added for SPM-11, the Software Performance Monitor.

- RMS-11 V2.0 is incorporated into the RSX-11M-PLUS distribution kit. There is no longer a separate RMS-11 distribution disk or tape, nor is there separate installation documentation.

- Pregenerated RSX-11M-PLUS systems are now available in both RL02 and RC25 disk kits. The following is a list of some of the changes that have been made to the RL02 pregenerated kit since the last release of RSX-11M-PLUS:

  - The pregenerated disk kits each contain two pregenerated Executives, one that includes I- and D-space support and one that does not. Included on each disk is an indirect command file you can use to increase available space on your system disk by deleting the Executive that you do not need.

  - The pregenerated kits feature a special, easily edited STARTUP.CMD.

  - Support for the RA60, RA80, RA81, RC25, RD51, RX50, TU80, and TSV05 has been added to the pregenerated kits.

- Certain tasks have been removed from the pregenerated system kits to increase available pool space.

See Chapter 5, Pregenerated RSX-11M-PLUS Kits, of the RSX-11M-PLUS System Generation and Installation Guide for information on these and other changes.

1.2.2 RSX-11M/M-PLUS MCR Operations Manual (U)

The following are new MCR features for V2.1:

- MCR commands now support the following new disk devices: RA81, RA60, RC25, RD51, RX50, TSV05, and TU80.
- The /REMOTE and /SPEED keywords for the MCR SET command support new terminal speeds and speed combinations for the DHV11 multiplexer. The new speeds are 75, 110, 134.5, 150, 300, 600, 1200, 1800, 2000, 2400, 4800, 9600, and 19200. In addition, these keywords support new speeds for the DH11, DZ11, and DZV11.
- The /TERM keyword for the MCR SET command now supports the LA50. See Section 1.5.1 of the Release Notes.
- The MCR OPEN command has two new keywords to support I- and D-space tasks. The keywords are as follows:

  /TASKI Accesses and prints the contents of addresses in the instruction space portion of an I- and D-space task.

  /TASKD Accesses and prints the contents of addresses in the data space portion of an I- and D-space task.

- The following new error message has been added for the REDIRECT command: "TT redirect error." This message means that an attempt was made to redirect TTn: to the null device.

1.2.3 RSX-11M/M-PLUS Command Language Manual (R)

This revision of the RSX-11M-PLUS Command Language Manual incorporates the following changes and additions:

- COBOL-81, rather than COBOL-11, is now the default compiler.

  COBOL[/C81] Invokes the COBOL-81 compiler to perform the compilation (default).

  COBOL/C11 Invokes the COBOL-11 compiler to perform the compilation.

  DCL command line for COBOL-81:

  ```
  COBOL[/C81] [/qualifiers[s]] filespec [/qualifier[s]]
  ```

  The DCL qualifiers and their MCR translation follow. A (D) indicates the default.

	DCL Qualifier	MCR Syntax
-	/OBJECT[=file] (D)	COBOL file,[list],[diag]=source
	/NOOBJECT	COBOL ,[list],[diag]=source

Specifies whether an object file is created.

-	/LIST[=file]	COBOL [obj],file,[diag]=source
	/NOLIST (D)	COBOL [obj],,[diag]=source

Specifies whether a compile listing is created.

-	/DIAGNOSTICS[=file]	COBOL [obj],[list],file=source
	/NODIAGNOSTICS (D)	COBOL [obj],[list]=source

Specifies whether a diagnostics file is created.

-	/OVERLAY_DESCRIPTION	/BLD
	/NOOVERLAY_DESCRIPTION (D)	/-BLD

Specifies whether an indirect command file (.CMD) and an Overlay Description Language (.ODL) file are produced. If you use the LINK/C81 command to task-build a program, do not use this qualifier. LINK/C81 also produces a .CMD and an .ODL file.

-	/CODE:CIS	/CIS
	/CODE:NOCIS (D)	/-CIS

Specifies whether the compiler should use CIS instructions in the object code it produces.

-	/CROSS_REFERENCE	/CRF
	/NOCROSS_REFERENCE (D)	/-CRF

Specifies whether cross-reference tables should be appended to the compiler listing.

-	/ANSI_FORMAT	/CVF
	/NOANSI_FORMAT (D)	/-CVF

Indicates whether the source program is in conventional ANSI format or DIGITAL's Terminal format.

-	/DEBUG	/DEB
	/NODEBUG (D)	/-DEB

Specifies whether the symbolic debugger is used. If this qualifier is used, the LINK/C81/DEBUG command must also be used to link the file.

-	/SHOW	/MAP
	/SHOW:MAP	/MAP
	/SHOW:NOMAP (D)	/-MAP
	/NOSHOW	/-MAP

Indicates whether the Data Division and Procedure Division offset maps are to be included in the list file.

-	/TRUNCATE	/TRU
	/NOTRUNCATE (D)	/-TRU

Indicates whether the compiler is to perform decimal truncation on the values of COMP data items.

- /CHECK /PER/BOU
  /NOCHECK (D) /-PER/-BOU
  /CHECK:ALL /PER/BOU
  /CHECK:NONE /-PER/-BOU
  /CHECK:BOUNDS /BOU
  /CHECK:NOBOUNDS /-BOU
  /CHECK:PERFORM /PER
  /CHECK:NOPERFORM /-PER

  Deals with run-time error checking. /CHECK enables subscript (index) range checking and nested PERFORM checking. /NOCHECK suppresses both of these. /CHECK:NOPERFORM and /CHECK:NOBOUNDS suppress each check individually.

- /NAMES:aa /KER:aa

  Tells the compiler to use the two alphanumeric characters specified as the PSECT kernel for this program.

- /WARNINGS (D) /INF
  /NOWARNINGS /-INF
  /WARNINGS:INFORMATIONAL /INF
  /WARNINGS:NOINFORMATIONAL /-INF

  Specifies whether the compiler should issue informational diagnostics during the compilation.

- /SKELETON (D) <no translation>
  /NOSKELETON /-SKL

  Specifies whether a skeleton Overlay Description Language (.SKL) file is created.

- /SUBPROGRAM /SUB
  /NOSUBPROGRAM (D) <no translation>

  Indicates whether this program is a subprogram.

- /TEMPORARY:device /TMP:device

  Tells the compiler to store its temporary work files on the specified device.

- The LINK/C81 command links COBOL-81 object files to produce a task image. This command will not work with versions of COBOL-81 prior to V2.0.

  DCL LINK/C81 command line:

  ```
  LINK/C81[/qualifier(s)] [filespec(,s)]

  LINK/COBOL[/qualifier(s)] [filespec(,s)]
  ```

  The new DCL qualifiers for this command follow.

  - /[NO]FMS
    /FMS:NORESIDENT

    The /FMS qualifier causes LINK/C81 to include Forms Management Services (FMS) library support in your task image. You must use this qualifier if you call FMS routines in your program. /NOFMS tells LINK/81 not to include FMS support: this is the default.

The /FMS:NORESIDENT qualifier causes LINK/C81 to include support for a non-memory-resident FMS library in your task image.

- /OTS:[NO]RESIDENT

  The /OTS:RESIDENT qualifier includes memory-resident OTS in your task image.

- /RMS:[NO]RESIDENT
  /NORMS:NORESIDENT

  The /RMS:RESIDENT qualifier creates a reference to the shared RMS-11 memory-resident library, RMSRES. This library includes input and output support for sequential, indexed, and relative file organizations.

- /[NO]MAP

  The /MAP qualifier causes LINK/C81 to produce a Task Builder map file with the file type .MAP. /NOMAP is the default.

- /[NO]DEBUG

  The /DEBUG qualifier tells LINK/C81 to include the COBOL-81 Symbolic Debugger in your task image. You must have assembled the object module with the /DEBUG switch. /NODEBUG is the default.

- New commands have been added to invoke the CORAL 66 compiler. These are:

  - /[NO]CHECK

    The /CHECK qualifier specifies that the code generated will check that all subscripts used in array and switch references are within bounds. /NOCHECK is the default.

  - /CODE:arg
    EIS
    FIS
    FPP
    PIC
    (EIS,PIC)
    (FIS,PIC)
    (FPP,PIC)

    The /CODE qualifier allows you to choose at compile time the instruction set for which the code is generated, regardless of the machine used to compile the source program.

  - /[NO]EXTEND_SOURCE

    The /NOEXTEND_SOURCE qualifier causes the compiler to process only the first 72 characters of source input per line. The default is /EXTEND_SOURCE, meaning all characters in an input line are processed.

  - /[NO]LIST

    The /LIST qualifier allows you to specify whether a compiler listing should be generated. The default is /NOLIST, meaning no compiler listing is generated.

- /[NO]MACHINE_CODE

  The /MACHINE_CODE qualifier specifies that you want the compiler listing to include a representation of the machine code generated. The default is /NOMACHINE_CODE.

- /NAMES:x

  The /NAMES qualifier causes the first character of compiler-generated program-section names to be a character, x, in order to make them unique for this compilation.

- /[NO]OBJECT[:filespec]

  This qualifier allows you to specify whether or not the compiler must generate an object module. The default is /OBJECT, which generates an object module.

- /[NO]OPTIMIZE[[:LEVEL]:n]
  /OPTIMIZE=n

  The /OPTIMIZE qualifier allows you to optimize the function of CORAL 66 source programs. The default is /OPTIMIZE.

- /READ_ONLY[:arg]
  ALL
  NONE
  PURE_DATA

  The READ_ONLY qualifier allows you to alter the READ-ONLY or READ-WRITE attributes of program sections.

- /[NO]SHOW[:(arg[,s])]
  ALL
  [NO]EXPANSION
  NONE
  [NO]OVERRIDE
  [NO]SOURCE
  [NO]STATISTICS
  [NO]SYMBOLS

  The /SHOW and /NOSHOW qualifiers specify elements you want included in or excluded from the listing file.

- /[NO]STANDARD

  The /STANDARD qualifier highlights non-IECCA keywords in the listing as warning messages. /NOSTANDARD suppresses this function; this is the default.

- /TEST[:n]

  The /TEST qualifier allows you to retain or omit declarations and statements from a particular compilation without editing the source text.

- /[NO]TRACEBACK

  The /TRACEBACK qualifier generates additional code to print source-file and line-number information in run-time error messages. /NOTRACEBACK is the default.

- /WIDTH:n

  The /WIDTH:n qualifier allows you to set the listing file to a specific width in the decimal range 8 through 132. /WIDTH:132 is the default.

- EDT is now the default editor.

- The /CHECKPOINT_FILE and /WIDTH qualifiers for the SET DEVICE command now accept decimal numbers by default.

- The /HFILL and /VFILL qualifiers for the SET TERMINAL and SHOW TERMINAL commands have been eliminated. However, the existing /CRFILL and /LFFILL qualifiers perform the same functions, respectively, as the two removed qualifiers.

- A new SHOW PROCESSOR command has been added to DCL:

  ```
  SHOW processortype processorname[/qualifier]
       BATCH
       CARD_READER
       INPUT
       PRINTER
       PROCESSOR
  ```

  The SHOW PROCESSOR command displays information about processors, card readers, batch processors, and other devices under the control of the Queue Manager.

  CARD_READER and INPUT are synonyms. PRINTER refers to all nonbatch output processors. BATCH refers to all batch processors.

  A simple SHOW PROCESSOR command displays all processors on the system.

- A new qualifier, /NOWARNINGS, has been added to the INITIALIZE/QUEUE command. The command syntax is:

  ```
  INITIALIZE/QUEUE queuename/NOWARNINGS
  ```

  This qualifier inhibits error messages. If you do not use this qualifier, you receive error messages by default.

- Because multivolume disk backup and restore operations have been added to V2.1, the BACKUP command has two new qualifiers as well as new functions for several existing qualifiers. These qualifiers are as follows:

  - /APPEND

    Appends new data to a tape or disk that already has one or more backup sets.

  - /DIRECTORY

    Creates UFDs (if they do not already exist) on a mounted output volume, then copies into these UFDs the files from the analogous UFD on the input volume.

- /IMAGE:arg
  SAVE
  RESTORE

  Specifies that you want to do a multivolume disk-to-disk backup or restore operation.

- /SAVE_SET

  Specifies the name of the backup set to be placed on tape or disk. For tapes and disks mounted foreign, the default name is the volume name for the disk being backed up. For mounted input or output disks during an image backup or restore operation, you can specify the full backup set name with SAVE_SET.

- BASIC-PLUS-2 is now the default BASIC compiler. The compiler can create Task Builder Command and Overlay Descripton Language files and a listing file, in addition to the object module file. (Note that BASIC/B11 runs BASIC-11.)

- FORTRAN-77 replaces FORTRAN IV-PLUS. (DCL supports both FORTRAN IV and FORTRAN-77.) The following new qualifier has been added to FORTRAN-77:

  - /[NO]STANDARD:arg
    ALL
    NONE
    SOURCE
    SYNTAX

    Directs the compiler to look in your source code for extensions to ANSI standard FORTRAN at the full language level.

1.2.4 RSX-11M/M-PLUS Utilities Manual (U)

Following are additions and changes for the Backup and Restore Utility (BRU) and the Task/File Patch Program (ZAP).

Backup and Restore Utility (BRU)

- BRU now does multivolume backups for disks. You can copy a large disk onto several smaller disks or several smaller disks onto a mounted large disk. However, the disks to which you copy are in BRU format, not Files-11. This means that, before you can read what is saved on the disks, you must restore them with BRU. This process is similar to magnetic tape backup and restore operations.

  The following new qualifier has been added to BRU to perform multivolume disk backups.

  - /IMAGE:SAVE
    :RESTORE

    Specifies that you want to do a multiple disk-to-disk backup or restore operation. If you are doing a backup operation, you must specify the SAVE option on the command line. If you are doing a restore operation, you must specify the RESTORE option on the command line.

  Because of the addition of multivolume disk backups, the meanings of the following BRU qualifiers have changed.

- /APPEND

  Directs BRU to append a backup set from the input disk volume to the last backup set on the output tape, or on the output disk if you are using the /IMAGE qualifier.

- /BACKUP_SET:name

  Specifies the name of the backup set to be placed on tape or disk. For a mounted input or output disk during an image backup or restore operation, you can specify the full backup set file name with the /BACKUP_SET qualifier.

- /DIRECTORY

  Lists at your terminal the backup set names or files on the specified tape or disk volume.

- /INITIALIZE

  Specifies that you want to initialize the output disk during a tape-to-disk or disk-to-disk operation.

- /INVOLUME

  Specifies the volume label of the input disk.

- /MOUNTED

  Allows you to back up files from a disk that is mounted as a Files-11 volume.

- /NEW_VERSION

  Resolves file specification conflicts, which occur during either backup or restore operations to a mounted disk.

- /SUPERSEDE

  Specifies that when file specifications on the mounted output volume are identical to the file specifications on the input volume, the file on the output volume is deleted and replaced with the file from the input volume.

- BRU has a new sort algorithm, which increases the speed of the sorting procedure.

- If you use the /MOUNTED qualifier when the input device is a tape, BRU now issues a syntax error message. The purpose of this error message is to clarify to the user that tapes cannot be mounted as Files-11 volumes. Previously, BRU ignored this qualifier if the user issued it for a tape input device.

- New examples have been added to BRU. They are as follows:

  - Disk-to-disk multivolume backup operations.

  - Disk-to-disk multivolume restore operations.

  - Disk-to-disk multivolume backup and append operations.

  - Disk-to-disk multivolume restore operations for restoring appended backup sets.

- New error messages for multivolume backup and restore operations have been added to the BRU chapter.
- BRU now supports the following additional devices.

Type	Mnemonic
TSV05/TU80 magnetic tape	MS
ML11 electronic memory	EM
RA60/RA81/RC25/RD51/RX50 disk	DU

- Changes have been added to booting stand-alone BRU for RSX-11M-PLUS. The stand-alone BRU system, BRUSYS, now requires 124K words of memory.
- On RSX-11M-PLUS, you now can copy a DB:-, DM:-, EM:-, DR:-, or DU:-type disk to any other of these controller types without having to re-SAVE the system, because the boot block for these devices is common. You now only have to use the following command to write the correct boot block on the output disk:

```
MCR SAV /WB
```

Task/File Patch Program (ZAP)

- ZAP has been enhanced to support user data space tasks and multiuser tasks. In addition, ZAP displays more information on the overlay structure of a task. See the new section in Chapter 15 of the RSX-11M/M-PLUS Utilities Manual, which describes the use of the /LI qualifier with both I- and D-space tasks and Multiuser Task Image files.

1.2.5 RSX-11M/M-PLUS I/O Drivers Reference Manual (U)

The following new devices are supported by RSX-11M-PLUS V2.1.

Supported disks:

Type	Mnemonic
RA81 fixed media disk	DU
RA60 removable media disk	DU
RC25 fixed media/removable cartridge disk	DU
RD51 fixed media disk	DU
RX50 floppy disk	DU

Supported magnetic tapes:

Type	Mnemonic
TU80 magnetic tape	MS
TSV05 magnetic tape	MS

Supported line printers:

Type	Mnemonic
LN01 line printer	LP
LP07 line printer	LP
LP26 line printer	LP
LP27 line printer	LP

Supported terminal devices:

Type	Mnemonic
LQP02 letter quality printer	TT
LA50 personal printer	TT
LA100 letter printer	TT
DHV11 multiplexer	YH

1.2.6 RSX-11M/11M-PLUS Task Builder Manual (R)

Following are the major technical changes documented for the Task Builder:

- New Options
  - DSPPAT--Allows object-level patching of a conventional task or the D-space part of an I- and D-space task.
- Changed Options
  - ABSPAT--Allows object-level patching of a conventional task or the I-space part of an I- and D-space task.
  - COMMON--Causes the common to be mapped with D-space APRs. Therefore, for I- and D-space tasks, the common can contain data only.
  - EXTTSK--Extends the D-space portion of an I- and D-space task. Because a library is mapped with both I-space and D-space APRs, extending the D-space of an I- and D-space task may result in the unmapping of the library's D-space APRs. This unmapping causes the library to be mapped in I-space only.
  - LIBR--Causes the library to be mapped with both I-space and D-space APRs when it is linked to an I- and D-space task.
  - RESCOM--Causes the common to be mapped with D-space APRs. Therefore, for I- and D-space tasks, the common can contain data only.
  - RESLIB--Causes the library to be mapped with both I-space and D-space APRs when it is linked to an I- and D-space task.
- New Error Message

  Module module-name contains incompatible autoload vectors

- Changed Error Message

  Original: Lookup failure resident library file

  Revised: Lookup failure resident library file - filename.ext

- Miscellaneous Technical Changes

  - Autoload vectors for conventional tasks have changed. The call to $AUTO is now made indirectly through .NAUTO in the overlay impure area.

  - I- and D-space tasks may be overlaid by using either autoload or manual load.

  - Autoload vectors for I- and D-space tasks have a format different from those of conventional tasks. Also, autoload vectors for I- and D-space tasks contain an I-space part, located in the task's I-space, and a D-space part, located in the task's D-space.

  - You now can write an .ODL specification by applying certain rules to a virtual address space diagram.

  - Overlay run-time system routines have changed size from the previous release.

  - MACRO-11 and FORTRAN manual load calling sequences for overlays in I- and D-space tasks may not use asynchronous loading.

  - For versions of TKB that both support I- and D-space tasks and build libraries, TKB allocates autoload vectors in the root of the task only for those autoloadable entry points in the library referenced by the task.

  - I- and D-space tasks may link to commons, conventional libraries, and supervisor-mode libraries.

  - Loading I- and D-space tasks into memory requires two disk accesses. Also, if the segment contains both I-space and D-space, overlaid I- and D-space tasks may require two disk accesses for loading each segment.

  - Segment descriptors for I- and D-space tasks contain an extension for the D-space part.

  - Only one level of overlay is allowed in supervisor-mode libraries.

  - I- and D-space multiuser tasks are allowed. TKB uses four window blocks to map these tasks.

  - Internal Symbol Directory Records, along with their formats, are described in Appendix A. They consist of the following:

    - Type 1 records, generated by TKB and output to the .STB file

    - Type 2 records, generated by language processors

    - Type 3 records, created from type 2 records and output to the .STB file.

  - Type 4 records, written to the .STB file without modification.

- A new bit called LD$TYP distinguishes between a library and a common. This bit is located at the offset R$LFLG in the resident library name block data, which is described in Appendix B.
- The first library in a cluster may be overlaid and contain a non-null root.
- New Task Builder reserved symbols have been added to Appendix E.
- The Fast Task Builder supports the /EA switch and the TASK= option.
- The map format for an I- and D-space task shows both I- and D-space contributions to a segment and the disk blocks that contain data sections.
- Other, minor technical and editorial changes have been made also.

1.2.7 RSX-11M/M-PLUS System Management Guide (U)

The following are changes or additions to the Virtual Monitor Console Routine (VMR).

- The /REMOTE and /SPEED keywords for the SET command support new terminal speeds for the following devices: DH11, DHV11, DZ11, and DZV11.
- The SAVE command now supports the following new disk devices: RX50, RA60, RC25, TSV05, and TU80.
- The upper limit for SET/SWPR is now 127(10).
- The SET command supports the LA50. (Note that this support is not documented in the VMR chapter.)
- The following new error message has been added for the REDIRECT command: "TT redirect error." This message means that an attempt was made to redirect TTn: to the null device.

1.2.8 RSX-11M/M-PLUS Error Logging Manual (R)

The following are changes or additions to Error Logging:

- Error Logging now allows hard and soft limits to be reached independently. Previously, reaching one of the limits would disable logging of either kind of error on that device. Now, reaching the soft limit does not affect the logging of hard errors and vice versa.
- Device timeouts are now logged as hard errors if unrecoverable and as soft errors if recoverable.
- When generating a report, RPT looks first for LX:[1,6]ERRLOG.ULB. LX: is a new pseudo-device used by Error Logging. If it fails to find that file, it looks for LB:[1,6]ERRORLOG.ULB.

- There have been a number of minor changes in the Control File Language (CFL) component of the Error Logging System. Here is a list of the differences in CFL between RSX-11M-PLUS V2.1 and the previous release.

  - %CNV$xxx functions:

    The field_width parameter is now optional. This parameter interacts with the optional fill_character parameter to determine whether the resulting string is simply printed as is or is left- or right-justified. In RSX-11M-PLUS V2.0, the digits in the string were always right-justified and blank-filled if no fill_character was specified.

  - %LOK$LENGTH function:

    This function returns the length of the data in a packet or subpacket. The length word for the packet or subpacket is not considered part of the data and is not counted in determining the length value returned.

  - %LOK$BYTE, %LOK$WORD, %LOK$LONGWORD functions:

    The offset parameter is the offset within the data of the packet or subpacket at which the byte, word, or longword begins. The offset unit is always in bytes, with the first byte of data in the packet or subpacket being offset 0.

  - %STR$UPCASE function:

    STR$UPCASE accepts an ASCII string as a parameter and returns this string with all lowercase ASCII characters converted to uppercase.

  - WRITE and WRITE_GROUP statements:

    Because of overlay restrictions, the following operators and functions cannot be used in expressions in WRITE or WRITE_GROUP statements:

    single and double operand numeric operators
    the MATCH operator
    %COD$xxx functions
    %CTL$xxx functions
    %PKT$xxx functions
    %RPT$xxx functions
    %STR$xxx functions
    %TIM$xxx functions
    %USR$xxx functions

  - DYNAMIC_TABLE statement:

    This new statement is a synonym for the FILE statement. You should use this new statement in place of FILE in any new code.

  - FILE statement:

    The FILE statement will be removed in a future release. Please convert your code to use the DYNAMIC_TABLE statement instead of FILE.

- The RSX-11M/M-PLUS Error Logging Manual has two new chapters.
  - Chapter 4, Error Logging Control File Architecture, explains the control file modules in detail, including flow of program control, interfaces between modules, and module dispatching. A knowledgeable system programmer can use the information presented here to add user-written modules to the Error Logging System. The chapter includes extensively annotated examples of DIGITAL-supplied modules.
  - Chapter 5, Control File Language Guide, describes the Control File Language, which is used to write control-file modules.

1.2.9 RSX-11M-PLUS Mini-Reference (R)

This revision to the RSX-11M-PLUS Mini-Reference consists of a new section on RMS-11, as well as technical improvements to several other sections of the manual.

The RMS-11 section includes both procedures for invoking the following RMS-11 utilities and summaries of their commands and keywords.

- RMSDES (RMS-11 File Design Utility)
- RMSIFL (RMS-11 Indexed File Load Utility)
- RMSCNV (RMS-11 File Conversion Utility)
- RMSDSP (RMS-11 File Display Utility)
- RMSBCK (RMS-11 File Back-Up Utility)
- RMSRST (RMS-11 File Restoration Utility)

In addition, the RMS-11 section lists the RMS-11 completion codes and fatal error codes.

The sections on Executive Directives, Directive Error Codes, I/O Error Codes, Task Builder Switches and Options, DCL, MCR, Error Logging System, and some utilities have been updated to include new or corrected information.

1.3 NOTES ON SOFTWARE

The following sections describe the incompatibilities that exist between RSX-11M-PLUS V2.1 and the previous version, and provide miscellaneous information pertinent to the successful use of the software.

1.3.1 Installing Tasks with External Headers

The INSTALL command uses the /XHR keyword to specify whether a task is installed with an external header. You can change the effect of this keyword by selecting one of the options described below. The options determine the default action the system takes when a task is installed. To select an option, the system manager must modify the build command file for the INSTALL task (INSBLD.CMD).

Note that the options do not apply to tasks that were built with the /-XH switch specified. The headers for these tasks are always in pool when the tasks are installed, regardless of the option selected, because the /XHR keyword cannot override the /-XH switch.

The following options are available:

1. By default, every task is installed with an external header. You can override this option by specifying /XHR=NO.

   This option is provided for users who want their system always to install tasks with external headers. The option also prevents users from unnecessarily or accidentally using headers in pool.

   For existing RSX-11M-PLUS sites, selecting this option allows the use of external headers without having to rebuild the nonprivileged application system.

2. By default, the headers for privileged tasks are in pool, and the headers for nonprivileged tasks are external. This option can be overridden when a task is built or installed.

   This option is provided for users who have user-written privileged code that needs to be converted before it can execute with an external header.

   The application system does not have to be rebuilt to use external headers.

3. By default, the headers for all tasks are in pool. You can override this option when the task is built (/XH) or when the task is installed (/XHR=YES).

   This option is provided for users who have user-written ACPs or privileged code that examines the header of another task.

   If this option is selected, a task must be either rebuilt or installed with an external header (TKB /XH or INS /XHR=YES) before it can execute with an external header. Because any user can specify the /XH switch, this option is recommended for friendly environments. (Only privileged users can specify the /XHR keyword.)

4. By default, the headers for all tasks are in pool. The option can be overridden by a privileged user by specifying /XHR=YES when the task is installed .

   This option is similar to option 3, but it cannot be overridden by the Task Builder's /XH switch. The option gives the system manager more control over the environment and must be selected for those systems that cannot allow any code, including nonprivileged code, to execute with external headers.

5. By default, the headers for all tasks are in pool. This option cannot be overridden.

   Selecting this option is not recommended. The effect of the option is similar to choosing a nonstandard Executive during system generation, so that support for external headers is not included.

This option must be selected, however, for those systems that cannot allow any code, including nonprivileged code, to execute with external headers.

To select an option, INSBLD.CMD must include the following command line:

```
GBLPAT = INSROT:$HDDEF:n
```

where n is the number of the desired option. The default option is 1.

After the command file has been modified, the INSTALL task must be rebuilt and then reinstalled in the system with VMR.

1.3.2 TDX (Catch-All Task)

Support for TDX, the catch-all task, is provided with RSX-11M-PLUS systems. The primary purpose of the catch-all facility is to give you the means to run uninstalled tasks.

Any task installed with the task name ...CA. is treated as a catch-all task. If the Monitor Console Routine (MCR) receives an unrecognized command, it searches for a task with that name and passes the command line to this task.

To use TDX as the catch-all task for your system, install it as follows:

```
>INS $TDX/TASK=...CA.
```

TDX checks the typed command against its list of commands. If the commands match, TDX translates the command into a legal MCR command. These commands and their MCR translations are:

Command	Translation	Meaning
ATS	ACT /ALL	Display the names of all active tasks in the system.
ATS ttnn:	ACT /TERM=ttnn:	Display the names of all active tasks on the specified terminal.
CHD	SET /UIC	Display the current default UIC for terminal TI:.
CHD g m	SET /UIC=[g,m]	Change the default UFD to the UFD specified.
CLR		Clear the issuing terminal's screen and set cursor to 0,0. Returns exit status of EX$SUC if terminal is a CRT, and EX$WAR if it is not.
CRE file	PIP file=TI:	Create a new file without invoking an editor.

Command	Translation	Meaning
CVT val		Evaluate an arithmetic expression, convert that expression into different formats, and display all the formats on your terminal. CVT accepts input in octal (nnn or nn,nn) or decimal (nnn. or nn.,nn.) words or bytes, hexadecimal numbers ($nnnn), radix-50 (%ccc) or ASCII ('c or "cc) characters, or arithmetic expressions using +,-,/,* and < >.
DEL file(s)	PIP file(s)/DE	Delete the specified file(s).
DIR [file[s]]	PIP [file[s]]/LI	Display a directory listing at the terminal.
DLG	DEV /LOG	Display information about all the logged-in terminals on the system.
DLN	NCP SHOW KNOWN NODES	Display all known DECnet nodes.
FRE	PIP /FR	Display the amount of available space on SY:, the largest contiguous space on SY:, the number of available file headers, and the number of file headers used.
FRE ddu:	PIP ddu:/FR	Display the same information as FRE for a specified device.
PUR file(s)	PIP file(s)/PU	Delete all but the latest version of a file.
SHQ	SHOW QUEUE ALL	Display information about all entries in all print queues.
SYS	SET /SYSUIC	Display the current system UIC.
TDX		Display the current version of TDX.
TYP file(s)	PIP TI:=file(s)	Print files on your terminal.

You may want to add other commands to TDX by modifying the source file in [24,10]TDX.MAC. The routines for the commands are at the end of the source file and serve as examples for user-tailored routines. To reassemble the source file after making additions, use the TDXASM.CMD file located in [24,24]. To rebuild TDX, follow the procedure in the Building Nonprivileged Tasks section of the RSX-11M-PLUS System Generation and Installation Guide.

If the user's typed command does not match any of TDX's commands, TDX attempts to issue one of the following two MCR command options. In the examples, XXX represents the first three characters of your command.

The option TDX uses depends on whether the logical devices ZZ1: or ZZ2: exist on your system. You can assign one of these devices in your LOGIN.CMD file. Note that the presence of the ZZ1: assignment masks the presence of the ZZ2: assignment. If no assignment is made, TDX will not exercise either option.

1.

```
MCR> RUN $XXX/TASK=XXXTTN/CMD="params..."
```

This option installs, runs, and then removes a task. "Params..." represents the command line that you enter. Your command line cannot exceed 39 characters, because TDX adds other characters to this line. If you want this option, place the following in your LOGIN.CMD file:

```
ASN SY:=ZZ1:
```

2.

```
MCR> @SY:[loginuic]XXX.CMD or
MCR> @LB:[libuic]XXX.CMD or
MCR> @SY:[loginuic]CATCHALL.CMD or
MCR> @LB:[libuic]CATCHALL.CMD
```

For this option, TDX searches for one of the @[...]XXX.CMD indirect command files, in the order indicated. As soon as TDX locates one of these files, the search stops. You can design the indirect command file to perform a variety of functions, such as installing tasks, providing HELP, or issuing error messages.

If you want this option, place the following in your LOGIN.CMD file:

```
ASN SY:=ZZ2:
```

Note that RSX-11M-PLUS V2.1 contains a sample indirect command file, [24,24]CATCHALL.CMD.

The ZZn option may be controlled for all users by placing the following command in the system startup file:

```
ASN SY:=ZZn:/GBL
```

where n equals 1 or 2, depending upon which ZZn option you prefer.

TDX issues the following message if the user's command does not match a TDX command or if TDX cannot locate the necessary task.

```
MCR -- Task not in system
```

You may wish to install some tasks permanently, rather than invoking TDX to install them on an as-needed basis. It is advisable to permanently install:

- Tasks requiring a larger increment than the default (see the INSTALL command in the MCR Operations Manual)
- Frequently used tasks
- Tasks requiring commands longer than 39 characters

1.3.3 Restriction for DCL as the Catch-All Task

If your command line interpreter is set to MCR and DCL is installed as the catch-all task (...CA.), do not use the DCL INITIALIZE/UPDATE command to modify your disk parameters. When you enter this command, MCR interprets the INI portion of the command to mean INITIALIZE VOLUME, and consequently destroys all existing files on your disk. As alternatives, either use the MCR HOME command or set your command line interpreter to DCL before using the INITIALIZE/UPDATE command.

1.3.4 Changes in Placement of HELP Files

For all V2.1 distribution kits, the HELP files are now placed on the second, rather than the first, BRU backup set for magtape kits or disk volume for RK07 kits. The transfer of these HELP files to your system disk occurs without any action required on your part.

For the case of the RK07 distribution kit, the BASTART.CMD file has been modified. This command file now invokes PIP in order to copy the HELP files from the second RK07 disk to your system disk.

1.3.5 Power-Fail Recovery

A disk driver is called at its power-fail recovery entry point when a power-fail occurs. This recovery routine gives a disk sufficient time to spin back up and be ready to accept I/O requests.

There are three possible recovery procedures:

1. If a device is busy prior to the power-fail and the requested I/O has not completed, the routine times out and checks the device status until the device is ready for I/O operations to resume.
2. If a device is not busy prior to the power-fail and receives an I/O request after the power-fail, the routine times out and checks the device status until the device is ready for I/O operations to resume.
3. If a device is not busy prior to the power-fail and there are no pending I/O requests, operations resume with no special handling by the recovery routine.

Note in the first two cases that, if the maximum time-out count is reached before the disk spins back up and is ready, an unsuccessful I/O completion code is returned.

1.3.6 Changes to Code and Data Structures for Privileged Tasks

The changes documented in this section were new for RSX-11M-PLUS V2.0 rather than for V2.1. This information is included here because it has not yet been incorporated into the documentation set.

Many of the changes made to RSX-11M-PLUS V2.0 may affect the upgrading of privileged tasks and drivers that were written for a previous release. The following sections describe the most significant changes to code and data structures since the previous release. Note that nonprivileged tasks should not be affected by any of these changes.

1.3.6.1 Changes Made to Executive Code

- The module IOSUB.MAC was divided into four modules: IOSUB, EXESB, MEMAP, and MDSUB. The last three are new.

- The module DREIF now makes special checks on send packets queued to a task to determine whether the UCB address in the packet is that of a virtual terminal. The purpose is to allow data to be sent to a slave task whose TI: is a virtual terminal. If a user's task receives specially formatted send packets, then the slave task should be built with the /-SE switch. This sets the T3.NSD bit that prevents the new check from interpreting the UCB address field in the packet.

- For directives that take a TCB address in R0 as input, if R0 is greater than 140000, the TCB is mapped through APR 6.

- The Get Partition Parameters directive now returns the real virtual starting address of a privileged task's task region instead of 0 as in V1.0.

- Because of the implementation of user I/D-space, the definitions of the user mapping register symbols (such as UISAR0) may point to either of the instruction (UINARx) or the data (UDSARx) registers. A similar situation exists with the Kernel mapping register symbols. Privileged code must not locally define these symbols or assume they refer to a particular mapping register.

- Some Executive routines that were not previously in the Directive Partition may have been moved to this partition for V2.0. Thus, privileged code that calls Executive entry points should be checked to ensure that the entry points are still accessible.

- Drivers that address check their own buffers will have to change $ACHKW/$ACHKB/$ACHR0 to $CKBFW/$CKBFB/$CKBFR/$CKBFI. These new address check routines correctly maintain the partition I/O count.

1.3.6.2 Changes Made to Executive Data Structures

- The Task Control Block (TCB) has undergone many changes. The present format can be found in an appendix of the RSX-11M/M-PLUS Crash Dump Analyzer Reference Manual.

- The Partition Control Block (PCB) has also been changed. The new format can also be found in the RSX-11M/M-PLUS Crash Dump Analyzer Reference Manual. Note in particular that drivers that set UC.QUE and relocate user buffers must be aware of the new offset P.IOC.

- To enhance buffered I/O, the bit T2.TIO, which previously allowed for only one buffered I/O per task, has been changed to the byte T.TIO, which now allows a task to queue many buffered I/O requests simultaneously. This is one of the many changes made to the buffered I/O algorithm.

- Tasks that examine task headers must be altered, since the headers may no longer be in primary pool. Since many privileged tasks assign LUNs by loading the UCB address into the LUN table in the header, this may be a problem when upgrading to Version 2.0. Installing these tasks without an external header enables them to work until the tasks can be modified to support external headers. You can add this support by doing the following:

```
.IF DF  X$$HDR

MOV     $SAHDB,KISAR6           ; MAP TASK HEADER
MOV     $SAHPT,R2               ; GET ADDRESS OF HEADER

.IFF    ;X$$HDR

MOV     $HEADR,R2               ; GET ADDRESS OF HEADER

.ENDC   ;X$$HDR

<Header manipulations using R2>
```

NOTE

These instructions must be executed at system state and must not be mapped through APR6. Also the references to @$HEADR require no modification. This code is necessary only to address header offsets other than the saved stack pointer.

- If the task has an external header, P.HDR in the PCB is now 0.

- Prototype TCBs (for tasks whose name is in the form ...XXX) may now reside in secondary pool. Therefore, any task that scans the PCB list should be modified to check for P2.SEC in P.ST2. If set, the TCB corresponding to that PCB is in secondary pool. A task that scans the installed task list should also be modified to scan $PTCBL, which is the listhead of the secondary pool prototype TCBs.

- The word at offset O.STAT+2 of the offspring control block is now used to contain a secondary exit status, the TKTN abort code, which is filled in whenever the offspring exits. Previously, only the word at O.STAT was used. Note that more words in the exit status region of the OCB may be used in the future.

- The addition of alternate CLI support has changed how CLIs are associated with individual users.

Bits 1 - 4 in the word U.MUP of the terminal's UCB are now used to identify the CLI associated with that terminal. The bits form an offset into a table located at $CPTBL in the Executive module SYSCM. This module contains a pointer to the CPB (CLI Parser Block) for the CLI. Since only 4 bits are used to form this offset, the system is limited to 16 CLIs.

- Tasks that use A.TCB (not to be confused with A.TCBL) must check a new bit, AS.PRO in A.STAT. If this bit is set, then A.TCB contains the 32-word physical bias of the prototype TCB rather than the virtual address of the TCB in primary pool. This means that tasks will have to remap an APR in order to access the TCB where needed.

- The following Accounting data structures have changed: the User Account Block (UAB), the System Account Block (SAB), and the Mount Transaction. Tasks can differentiate between V1.0 and V2.0 by the length of the block.

1.3.6.3 Changes Made to I/O Data Structures

- Implementing the full-duplex virtual terminal driver required the rearrangement of its UCB and the addition of one word. The most important consideration is that the offset that contains the parent TCB address, U.PTCB, is now located after offset U.UIC.

- Some changes have been made that may affect user-written ACPs. The high bit of I.EFN determines whether a function is virtual (bit set) or logical (bit clear). Note that for a logical function, the header may not be in memory. The contents of I.LN2 is the address of the appropriate second LUN word. For a virtual function, the low bit of I.LN2 determines whether the header is locked down (bit set) or not locked down (bit clear). For virtual functions, I.LN2 contains the contents of the appropriate second LUN word (possibly using the OR operation with the low bit as described).

- The bit DV.MXD in offset U.CW1 of the UCB has been changed to DV.MSD. This bit is set for all mass storage devices. This bit must be turned on for all user-written drivers that want to support error logging or seek optimization.

  In addition, the UCB must include the offset U.UCBX, which follows offset U.CNT. U.UCBX is initially 0, but will contain the APR bias of the UCB extension when the device is configured on line. The UCB extension resides in secondary pool. Its format is documented in the RSX-11M-PLUS Guide to Writing an I/O Driver and also in an appendix of the RSX-11M/M-PLUS Crash Dump Analyzer Reference Manual. The SCB offsets dealing with error logging remain unchanged.

- Implementing settable remote answer speeds in the full- duplex terminal driver required the rearrangement of U.CW3 in the UCB.

1.3.7 Change to the Executive's Task Abort Code

The Executive's task abort code has been modified. If you install an Ancillary Control Processor (ACP) task with post-mortem dumps (PMDs) enabled and this task aborts, then the system will crash. A crash dump is produced, which shows the state of the I/O data structures at the time the ACP task aborted. By default, most DIGITAL-supplied ACPs are installed with post-mortem dumps enabled.

1.3.8 SYSGEN

The following notes concern RSX-11M-PLUS V2.1 SYSGEN:

- Cross references are no longer included in the Executive maps. However, the user can rebuild the Executive specifying a cross-reference listing. See the RSX-11M/M-PLUS Task Builder Manual.

- The procedure for generating a RSX-11M-PLUS V2.1 system on a V2.0 host is described in the RSX-11M-PLUS System Generation and Installation Guide. Note that you can use the V2.0 saved answer files for the V2.1 SYSGEN. Whether you choose to use the V2.0 saved answer files or not, SYSGEN asks whether you want software performance monitor (SPM-11) support.

- The way remote lines on a DH11 are assigned has changed in V2.1.

  In the past, when you chose remote lines on a DH11, the lines were assigned in groups of four. For example, if you specified two remote lines and six local lines, SYSGEN would generate four remote lines and six local lines.

  With this release, the remote lines are no longer assigned in groups of four. If you specify two remote lines and six local lines, SYSGEN will generate only two remote lines and six local lines.

  If you have DH11 remote lines, this change may cause fewer terminals to be generated into your system than in previous releases, and as a result, the terminal numbers (logical unit numbers of the terminals) may differ from those in previous releases.

1.3.9 Changes to the Command String Interpreter (CSI)

CSI was rewritten for V2.0 of RSX-11M-PLUS to incorporate new functionality. See the I/O Operations Reference Manual for a list of changes to CSI.

Two additional changes are:

- .CSI1 no longer references the module ODCVT.

- .CSI2 no longer references the module ODCVT and now references OD2CT.

These changes are normally not visible to you, but could be a problem for some overlaid tasks.

1.3.10 Task Builder Restrictions

The following two restrictions apply to the V2.1 Task Builder.

- You may experience problems if you use the V2.1 Task Builder (task version M40.10) with a SYSLIB other than the one supplied with V2.1. Changes and additions have been made to the overlay run-time system modules, required by overlaid I- and D-space tasks, that make it incompatible with previous versions of SYSLIB.OLB.

- Severe performance degradation can occur with cluster libraries if you have the following two conditions in your task:

  1. The first library is overlaid and has a null root.

  2. The first library called by the task is not the first library specified in the CLSTR option.

  The first library in a cluster of libraries may or may not be overlaid and may or may not have a null root. The remaining libraries in the cluster must be overlaid with a null root. The first library in a cluster is the one named first in the Task Builder cluster (CLSTR) option.

  If the first library in the CLSTR option is overlaid and has a null root, the overlay run-time system cannot distinguish the first library from the remaining libraries in the cluster. The Task Builder assumes that the first library called by the task is the first library: this first library is the one that the Task Builder consistently maps by default. Therefore, although the preceding two conditions do not cause errors, they may cause severe performance degradation because of excessive mapping and unmapping of libraries in the cluster.

  To avoid this problem, the library first called by the task should be the first library specified in the CLSTR option.

1.3.11 Modifications to the Task Builder (TKB)

Changes have been made to Task Builder options, as well as to overlays, libraries, I- and D-space tasks, and data structures. The following sections list these changes.

1.3.11.1 Options - The following TKB options have been modified:

ABSPAT Allows object-level patching of a conventional task or the I-space of an I- and D-space task.

COMMON The COMMON option causes the common to be mapped with D-space APRs. Therefore, the common can contain data only.

EXTTSK Extends the D-space portion of an I- and D-space task. Because libraries are mapped with both I-space APRs and D-space APRs, extending the D-space of I- and D-space tasks may cause unmapping of the library's D-space APRs. This upmapping of D-space APRs in turn may cause the library to be mapped in I-space only.

LIBR The LIBR option causes the library to be mapped with both I-space and D-space APRs.

RESCOM The RESCOM option causes the common to be mapped with D-space APRs. Therefore, the common can contain data only.

RESLIB The RESLIB option causes the library to be mapped with both I-space and D-space APRs.

1.3.11.2 **Overlays -**

- Overlay run-time system routines have changed size from the previous release.
- MACRO-11 and FORTRAN manual load calling sequences for overlays differ in I- and D-space tasks from those in conventional tasks.
- Only one level of overlay is allowed in supervisor-mode libraries.

1.3.11.3 **Libraries -**

- A new bit called LD$TYP distinguishes between a library or a common. It is located at the offset R$LFLG in the resident library name block data, which is described in Appendix B of the RSX-11M/M-PLUS Task Builder Manual.
- The first library in a cluster may be overlaid and contain a non-null root.
- Libraries created with older versions of TKB do not have the ISD records in the .STB file that newer versions of TKB use to create autoload vectors. Therefore, TKB must include autoload vectors from the .STB file for every point in the library. If you are using one of these older libraries and you are linking an I- and D-space task to it, TKB issues the following error message:

  Module module-name contains incompatible autoload vectors

  This message is issued because the .STB file contains conventional autoload vectors that are not usable by an I- and D-space task.

1.3.11.4 **I- and D-space Tasks -**

- I- and D-space multiuser tasks are allowed. TKB uses four window blocks to map these tasks.
- The map format for an I- and D-space task shows both I- and D-space contributions to a segment and the disk blocks that contain data sections.

1.3.11.5 **Data Structures -** Beginning with the V2.0 release, the Task Builder and the associated overlay run-time system incorporate new overlay functionality and modified data structures.

New modules are included in the system library to support cluster libraries (see the RSX-11M/M-PLUS Task Builder Manual for a complete description of this feature). In order to minimize the virtual memory impact of the new features, some changes were made in the older modules. The customer product that used the run-time routines may not build correctly or may malfunction unless minor changes are made. In particular, the following points are relevant:

- The run-time routines reference the register save routine .SAVR1 instead of .SAVRG. In some overlaid applications, this routine may have been explicitly loaded in an overlay (for example, as part of a tightly overlaid FCS-11). This reference now results in a "multiply defined global" message from TKB. The change is to remove explicit .SAVR1 references from overlay specifications in the .ODL file and to let TKB place this module in the root with the overlay run-time routines.

- The linkages between the various overlay run-time modules have been modified slightly to support the new library mapping; therefore, users who have developed debugging tools that "hook" themselves into the overlay system should refer to a copy of the source microfiche or the source kits.

- The .PSECT attributes of most SYSLIB modules has been changed from RW to RO to facilitate building multi-user tasks. The Task Builder automatically changes the data structures and issues multiply defined Program Section Diagnostic messages if a version of SYSLIB prior to V2.0 is used.

1.3.12 Restrictions for EDT

These restrictions apply to V2.0 of EDT, which is a layered product bundled on your RSX-11M-PLUS V2.1 distribution kit.

Before EDT can be run in keypad mode, your terminal must be set to lowercase. If the terminal is not set to lowercase, the terminal driver converts the escape sequences sent by the keypad keys into uppercase. EDT, therefore, cannot recognize the escape sequences.

Also, if you type CTRL/C while EDT is writing your output file (that is, after you type the EXIT command but before the CLI prompt appears), EDT stops writing the output file. No message is displayed to indicate that your output file has been only partially written.

1.3.13 MACRO-11

The maximum number of relocatable terms in a complex expression has been changed from the old maximum of 6. to 20. This change was made by increasing the maximum size of the .OBJ record that MACRO-11 produces from 42. bytes to 128. bytes.

You cannot verify that code generation is correct by comparing .OBJ files created by different versions of MACRO-11. The correct way to verify code generation is to task-build the .OBJ files, then to compare the .TSK image files.

Because the .OBJ files produced by the new version of MACRO-11 differ from those of previous versions, the user must recompute checksums using the PAT utility for any object patches assembled with the new version of MACRO-11.

See the Release Notes at the end of the PDP-11 MACRO-11 Language Reference Manual for documentation of changes to MACRO-11.

1.3.14 BRUSYS Restriction

On an RSX-11M-PLUS V2.1 stand-alone system, you should not run BRU and BAD simultaneously. Because BRU and BAD use common buffer space, running both tasks at the same time yields unpredictable results.

1.3.15 Device Drivers and User Tasks

The addition of new and enhanced device support to RSX-11M-PLUS V2.0 may cause problems for some user tasks in V2.1. The characteristics of the device drivers, and the procedures recommended for handling related problems, are described as follows:

- The UDA50 (DUDRV) dynamically updates the system data base to reflect the characteristics of the UDA50.

  Recommended Action: User tasks should issue a QIO Attach function before requesting the device's characteristics with the GET LUN directive.

- The TM11A/B device driver (MTDRV) dynamically updates the system data base to reflect the density characteristics of TE10/TU10.

  Recommended Action: User tasks should issue a QIO function using the function codes for the power-fail recovery procedure, as described in the RSX-11M/M-PLUS I/O Drivers Reference Manual, before requesting the device's density characteristics with the GET LUN directive.

- Users issuing QIOs directly to MSDRV must be aware of the following:

  1. Completion of an IO.RWD request occurs when the MS: device reaches BOT.

  2. When the MS: device changes status from off-line to on-line or vice versa, the MS: device inhibits further physical I/O operations. After such a change, the user must issue IO.RWD or IO.SMO requests that succeed before I/O can proceed.

  3. MS: read/write data transfer features are:

     - The data buffer starting address must be on a word boundary.

     - The data transfer size may be an odd or even byte count.

     - The user can swap odd and even data bytes by using the tape characteristic bit 1 of IO.SMO or IO.STC requests. When bit 1 is set to 0, no byte swap occurs; when it is set to 1, byte swap does occur. If you use byte swapping, it is recommended that the data buffer size be an even byte count.

- The RX02 device driver (DYDRV) dynamically updates the system data base to reflect the characteristics of the media in the RX02 drive.

  Recommended Action: User tasks should issue a QIO Sense Characteristics function before requesting the device's media characteristics with the GET LUN directive.

1.3.16 Changes to Task Checkpointing Characteristics

The /CP task-build switch has been substituted for the /AL task-build switch in the SYSGEN .BLD files for the following tasks: HRC, MCD, SYL, HEL, BYE, TKN, FCPLRG, SAV, SHF, SHU, INS, and PMT.

The reasons for this substitution are as follows: External task header support is automatically included in the Full-functionality Executive option, which most users select. Tasks with external headers can only be checkpointed to a system checkpoint file. The /AL switch, however, allocates space within the same task image; this space is never used by tasks with external headers. The /CP switch makes a task eligible for system checkpointing, thus eliminating the checkpoint space that the /AL switch had allocated within the task image.

1.3.17 Executive Debugging Tool (XDT)

Latent support has been included as a first step in providing the user with more detailed information in the event of a system crash. For RSX-11M-PLUS V2.1, the only change visible to the user is that XDT issues the error message EM:xxxxxx rather than IO:xxxxxx for some Executive faults or traps.

1.3.18 File Transfer Program (FLX)

The behavior of FLX in handling an error during a copy operation to an RT-11 device has changed. Previously, FLX allowed a partial file to be created on the output device when insufficient space existed on this device for the complete file. Then, if a file existed on the output disk with the same name as the newly created partial file, FLX deleted the preexisting file and substituted the partial file. This replacing of an existing file with a partial file resulted in a loss of information.

Now, if an error occurs during the copy operation, the partial file is not substituted for an existing file on the output disk. The partial file no longer is closed as a permanent entry, and the definition and contents of the existing file remain intact.

1.3.19 Incompatibility Problem with BRU Tapes

Because of BRU's new sort algorithm, V2.1 of BRU can read V2.0 BRU backup tapes, but V2.0 of BRU cannot read V2.1 BRU backup tapes.

1.3.20 Device Restrictions

The following device restrictions apply to RSX-11M-PLUS V2.1.

- MCR/DCL

  The SET and SHOW commands do not recognize the LQP02 letter-quality printer.

- File Transfer Program (FLX)

  The following devices do not support DOS-11 or RT-11 format:

  RC25
  RA60
  RA81
  RD51
  RX50

- Disk Volume Formatter (FMT)

  The following devices are not supported:

  RD51
  RX50
  RA60
  RA81
  RC25
  RP07

- Bad Block Replacement Control Task (RCT)

  The RX50 does not have an RCT area.

- Virtual Monitor Console Routine (VMR)

  - The SAVE command does not recognize the following devices:

    RA81
    RD51

  - The SET command does not recognize the following devices:

    LN01
    LP26
    LP27
    LP07
    LQP02

- Error Logging

  The Error Logging control files do not have device-specific information for the following devices:

  RD51
  RX50

- Crash Dump Analyzer (CDA)

  The following fixed disks cannot serve as CDA devices:

  RA80
  RA81
  RD51
  RC25 (nonremovable)

1.3.21 Building the COBOL-81 V1.1 or V1.2 Compiler

If you are installing COBOL-81 V1.1 or V1.2, you must answer the following dialogue prompts as indicated in order to build the COBOL-81 compiler correctly:

> Do you want to build the default compiler <YES>?

Do not take the default. You should type: NO

The next two questions in the dialogue concern CIS and RMS. Answer these as you wish.

The next question is:

> What should be the compiler task size in K words (legal range is from "26" to "32") <"32">?

The compiler actually builds at 1K greater than what you type. Therefore, type a number between 26 and 31 inclusive, and do not take the default. (Note that the number 25 is not accepted by the software.)

The next question asks whether you want to change the default of any of the compiler switches. You can either take the defaults or change them.

The entire COBOL-81 dialogue is documented in the COBOL-81 Installation Guide.

1.3.22 Queue Manager

The MCR /NM switch supresses messages from the Queue Manager (QMG) such as "Queue already exists."

In DCL, the equivalent switch is /NOWARNINGS.

Once a queue has been established, it will remain until eliminated, in spite of system crashes. But it is desirable to attempt to reestablish the standard print queues each time the system is bootstrapped, in case a queue has been inadvertantly eliminated. Issuing a QUE LP0:/CR command, though, yields the "Queue already exists" message from the QMG if the LP0 queue has not been eliminated, as is generally the case.

The /NM switch, therefore, is used in the prototype STARTUP.CMD supplied on the distribution kit to reduce the "clutter" that is printed on the console terminal when STARTUP.CMD is invoked.

1.3.23 Restriction to Error Logging

By moving the Error Logging routines into the Executive Common, the following restriction is introduced: if a driver of an Error Logging device calls the $CRPKT routine to create an Error Logging packet, the data address for the data subpacket must not be an address within the driver. Specifically, the address must not be mapped by APR 5, as this APR is used to map the common. Any user-written driver that performs such a function must allocate a piece of pool, fill in the appropriate information, and pass the pool address to the create packet routine.

1.3.24 User Account for the Warm-up Session

In UFD [200,1] on your target system disk are introductory files used with the warm-up session for new users presented in the Introduction to RSX-11M and RSX-11M-PLUS manual. When new users come onto the system, the system manager should have them use the account with the UIC [200,1] and the account name USER, which has been supplied with the distribution kit. New users can log into this account and follow through the entire warm-up session. See Section 4.5.4 of the RSX-11M-PLUS System Generation and Installation Guide and pp. 1-5 to 1-7 in the Introduction to RSX-11M and RSX-11M-PLUS.

Alternatively, the system manager can copy the introductory files to the new user's own account, then instruct the new user to delete these files when he or she has finished with the warm-up session.

1.4 REPORTING PROBLEMS AND APPLYING CORRECTIONS

RSX-11M-PLUS includes several features to use if you encounter questions or problems with your system or if DIGITAL provides temporary patches to improve its software performance.

Software Performance Reports (SPRs) allow you to report any problems directly to DIGITAL. The procedure for submitting an SPR is described in Section 1.4.1.

Sections 1.4.2 and 1.4.3 describe how to apply temporary software corrections to your system. You should use these instructions to apply manual patches that are given either in a DIGITAL response to an SPR or in a Software Dispatch article.

Note that any patches in the Software Dispatch are temporary and will be overriden by future DIGITAL Updates. An Update is a periodic software fix to your system, distributed in the form of machine-readable code. DIGITAL customers who are under warranty or who have purchased any level of software product service will receive the Software Dispatch and Updates.

The following sections explain the SPR process and the process for applying corrections to your system.

1.4.1 Software Performance Report

When you complete an SPR form, describe only one problem per form to simplify record keeping and facilitate a more rapid response.

An SPR can be used for:

- Software errors
- Documentation errors when the documentation comment form is not appropriate
- Inquiries
- Suggestions
- Follow-up on a previous SPR

An SPR cannot be used for:

- Software license and price policies
- Obvious hardware problems
- Logistical or clerical problems with kits or with Software Dispatch, such as blank media, or failure to receive the Software Dispatch
- Problems with user-written software

There are three categories of reports you can send:

1. Problem/Error

   This type of report should have a priority of one to five. (Answer is generated.)

2. Suggested Enhancement

   This type of report contains suggestions or queries that are brought to the development group's attention. (No answer is generated.)

3. Other

   This type of report contains queries and suggestions that are priority five. (Answer may be generated.)

The priority numbers are as follows:

1. Most production work cannot be run.
   - Major system functions are unusable.
   - You cannot boot the system.
   - Necessary peripherals cannot be used.
2. Some production work cannot be run.
   - Certain functions are unusable.
   - System performance has declined.
   - Installation does not have enough excess capacity.
3. All production work can be run with some user impact.
   - Significant manual intervention is required.
   - System performance has declined, but installation has excess capacity.
4. All production work can be run with no significant impact on user.
   - Problem can be patched or easily bypassed.
5. No system modifications are needed to return to normal production.
   - Suggestions, consultations, or errors in documentation.

Because problems are often difficult to reproduce with a different system configuration, please define as precisely as possible the state of the system when the problem occurred. Illustrate a general problem with specific examples. When you refer to a manual, specify the full title and section, table, or page number. Please supply the following information (in machine-readable form where applicable) when you report a problem:

1. CRASH - A copy of the Executive task-build map, output from the console terminal, the SYSGEN saved-answer file, the Executive STB file, and the crash dump. If the crash is reproducible, please accurately describe the details and supply a hard copy or user source code when necessary.

2. DRIVERS - Controller/device information, software options, error log output, copy of device registers, and a sample program.

3. UTILITIES - A copy of your terminal output, showing setup commands, before and after effects, and relevant file information.

4. TASK BUILDER - A copy of your terminal output command files, the task map, and a dump of the first few blocks of the task image.

5. FILE SYSTEM - Corrupted Volume: Output from Verify utility and dump of the volume. Improper results: Error code, file header dump, sample program.

6. ERROR LOG REPORT GENERATOR (RPT) - A copy of the report file generated by RPT, either on a hard copy listing or on machine-readable media.

If a failure occurs when you are running privileged, add-on software (for example, DECnet), try to reproduce the failure without the additional software. Indicate on the SPR the behavior of the system with and without the add-on software.

The SPR process takes time; therefore, if you have a critical problem, contact the local DIGITAL office. In the meantime, read the Software Dispatch for news on RSX-11M-PLUS, which includes changes and problems other people have found and solved.

1.4.2 Applying Corrections to Source Files

Interim changes to the Executive, MCR, and device drivers are made by creating correction files that are processed by the Source Language Input Program (SLP). SLP generates a new copy of the modules that contain the errors by applying the corrections to the source file on the distribution kit. (See the RSX-11M/M-PLUS Utilities Manual for complete information on SLP.)

After you have applied the corrections and obtained a new version of the file, DO NOT delete the original source file. Interim changes that may be distributed later are cumulative and depend on having the sources available.

1.4.2.1 **Updating an Executive Source Module** - To update an Executive source file (ABCDEF.MAC, for example), mount the disk on which you performed your SYSGEN and create a SLP correction file named ABCDEF.COR in the UFD [11,40]. (All instructions below assume that you are working on the disk on which you performed your SYSGEN.) Then, running under UIC [11,10], submit the correction file to SLP. For example, you could follow this sequence to update REQSB.MAC:

```
>SET /UIC=[11,40]
>EDI REQSB.COR
[CREATING NEW FILE]
INPUT
REQSB.MAC;2/AU/-BF=REQSB.MAC;1
.
.
.
*EX
[EXIT]

>SET /UIC=[11,10]
>SLP @[11,40]REQSB.COR
```

If the updated Executive module in your system is not a loadable driver, you must:

1. Assemble the new module, using the RSXMC.MAC file for the target system. For example:

   ```
   >SET /UIC=[11,24]
   >MAC REQSB,[11,34]REQSB/-SP=[1,1]EXEMC/ML,[11,10]RSXMC/PA:1,REQSB
   ```

2. Use LBR to replace the old version of the module in RSX11M.OLB on the target system. For example:

   ```
   >SET /UIC=[1,24]
   >LBR RSX11M/RP=[11,24]REQSB
   ```

3. Perform the following sections of SYSGEN:

   Building the Executive and Drivers
   Building the Privileged Tasks
   Creating the System Image File

If the modified file in your system is a loadable device driver (ZZDRV.MAC, for example), the updated module can be replaced without rebuilding the Executive. Assemble the updated module and replace the resulting object file in the RSX11M.OLB of your target system. For example, assuming that [11,10] contains the RSXMC.MAC file resulting from your system generation, do the following:

```
>SET /UIC=[11,24]
>MAC ZZDRV=[1,1]EXEMC/ML,[11,10]RSXMC/PA:1,ZZDRV
>SET /UIC=[1,24]
>LBR RSX11M/RP=[11,24]ZZDRV
```

Use [200,200]ZZDRVBLD.CMD to rebuild the driver. If necessary, copy ZZDRV.TSK and ZZDRV.STB into the UFD corresponding to the system UIC.

```
>ASN SY:=OU:
>TKB @[200,200]ZZDRVBLD
>SET /SYSUIC
SYSUIC=[g,m]
>SET /UIC=[g,m]
>PIP /NV=[1,54]ZZDRV.TSK,ZZDRV.STB
```

Use VMR to unload the old device driver and to load the new one. If the new driver is larger than the old one, it may not fit into the same locations as the old one. It may be necessary to unload and reload all of the loadable drivers in that partition to create enough room.

```
>VMR
ENTER FILENAME: RSX11M
VMR>UNL ZZ:
VMR>LOA ZZ:
VMR>^Z
>RUN $SHUTUP
```

Hardware boot the modified system.

1.4.2.2 **Updating an MCR Source Module** - To update the MCR source file SETOV.MAC, use the following procedure:

Create the SLP correction file [12,40]SETOV.COR and use it to update [12,10]SETOV.MAC. Assemble SETOV for the target system.

```
>SET /UIC=[12,40]
>EDI SETOV.COR
[CREATING NEW FILE]
INPUT
SETOV.MAC;2/AU/-BF=SETOV.MAC;1
.
.
.
*EX
[EXIT]

>SET /UIC=[12,10]
>SLP @[12,40]SETOV.COR
```

Assuming [11,10] contains the RSXMC.MAC file resulting from your target system generation, do the following:

```
>SET /UIC=[12,24]
>MAC SETOV=[1,1]EXEMC/ML,[11,10]RSXMC/PA:1,[12,10]SETOV
```

All of the Task Builder command files output a map to the logical device MP:. MP: must be assigned to NL: or another device to avoid a diagnostic error message from the Task Builder.

If it was necessary to modify the MCR Task Builder command file (MCRBLD.CMD) during the last system generation, it may now be necessary to repeat those changes. To rebuild the secondary portion of MCR (...MCR) and replace the module SETOV, use the following procedure:

```
>SET  /UIC=[1,24]
>LBR  MCR/RP/NOEP=[12,24]SETOV
>PIP  SETOV.OBJ;*/DE
>ASN  SY:=MP:
>ASN  SY:=IN:
>TKB  @MCRBLD
>VMR
ENTER FILENAME:  RSX11M
VMR>REM ...MCR
VMR>INS MCR
VMR>^Z
```

To rebuild the MCR dispatcher (MCR...) and replace the module MCRDIS, use the following procedure:

```
>SET  /UIC=[1,24]
>LBR  MCR/RP/NOEP=[12,24]MCRDIS
>PIP  MCRDIS.OBJ;*/DE
>ASN  SY:=MP:
>ASN  SY:=IN:
>TKB  @MCDBLD
>SET  /SYSUIC
SYSUIC=[g,m]
>SET /UIC=[g,m]
>PIP /NV=[1,54]MCD.TSK
>VMR
ENTER FILENAME:  RSX11M
VMR>REM MCR...
VMR>INS MCD/XHR=NO
VMR>^Z
>RUN $SHUTUP
```

There is only one procedure for replacing an external MCR task. It involves the following steps:

1. Create the SLP file, apply it, and create the object file.

2. Incorporate the updated module into the task's object library.

3. Rebuild the task and install it in the system, using MCR or VMR. Before using VMR, you must assign SY: and LB: to the disk containing the target system.

The following example replaces the module INSLB of the external MCR task Install:

```
>SET /UIC=[1,24]
>LBR INS/RP=[12,24]INSLB
```

If it was necessary to modify the external task's Task Builder command file during the last system generation, it may be necessary at this time to repeat those changes:

```
>SET  /UIC=[1,24]
>ASN SY:=MP:
>TKB @INSBLD
>SET /SYSUIC
SYSUIC=[g,m]
>SET /UIC=[g,m]
>PIP /NV=[1,54]INS.TSK
>VMR
ENTER FILENAME: RSX11M
VMR>REM ...INS
VMR>INS INS
VMR>^Z
>RUN $SHUTUP
```

Hardware boot the system.

1.4.2.3 **Updating a DCL Source Module** - The procedure is different for the DCL task. There are two DCL object libraries: DCLR.OLB for modules in the root segment of the DCL task, and DCLO.OLB for modules in DCL's overlay segments. (One module, COMMAND, has versions in both

the root and overlay libraries.) These libraries are located in UFD [1,24] of your distribution kit. If you are not sure whether a module belongs in the root or the overlay library, use LBR to scan the module names in the libraries. For instance, to view the module names in DCL0.OLB, enter the following command:

```
MCR:

LBR [1,24]DCL0/LI

DCL:

LIBRARY/LIST [1,24]DCL0
```

The modules you are most likely to need to modify are the DCL syntax tables which are in DCL0.OLB. (All DCL syntax tables are in overlay segments.) See Chapter 12 of the RSX-11M/M-PLUS System Management Guide for more information on the structure of the DCL task and on the DCL task-building process.

File DCL.CMD, in UFD [23,24] of your distribution kit, can be used either to assemble DCL overlay modules and to re-task-build DCL, or as a template for your own commands. The file contains its own instructions. Note that to use this command directly, you must copy various files into its UFD, and you must appropriately modify any UFDs that they reference. DCL.CMD also shows how to assemble the special module COMMAND.

The following process shows how to reassemble a DCL overlay module and re-task-build DCL. The process is similar for root modules; simply specify DCLR rather than DCL0. See DCL.CMD for details on rebuilding COMMAND.

```
MCR:

SET /UIC=[23,24]
MAC XXX=[11,10]RSXMC/PA:1,[23,10]DCLMAC/PA:1,XXX
SET /UIC=[1,24]
LBR DCL0/RP=[23,24]XXX
TKB @DCLBLD

DCL:

SET DEFAULT [23,34]
MACRO [11,10]RSXMC/PASS:1,[23,10]DCLMAC/PASS:1,XXX
SET DEFAULT [1,24]
LIBRARY/REPLACE DCL0 [23,24]XXX
LINK @DCLBLD
```

When you have built a new version of DCL.TSK, you must replace the old copy of DCL as a system CLI. This is done as follows: First, any terminals whose CLI is DCL must be logged off or set to another CLI. Then the following commands must be executed. (Note that these commands are MCR only.)

```
MCR:

CLI /ELIM=DCL
REM DCL
INS $DCL/CLI=YES
CLI /INIT=DCL
```

The REMove and INStall commands are also valid VMR commands and can be used to modify your system's image on disk. However, the CLI /INIT command is MCR only, and must be put in your system's startup file (if it is not there already).

1.4.2.4 **Applying Corrections to the File Systems (F11ACP)** - All F11ACP updates begin by creating a SLP correction file in UFD [13,40]. The following example patches a module called WTRN1.

Boot your system and log in to a privileged account:

```
>HELLO SYSMANAGER
PASSWORD:
```

If necessary, restore the required files from the distribution tape. (This example assumes that the files in [13,10] have been deleted, but the files in [1,24] have not.)

```
>UFD DB0:[13,10]
>UFD DB0:[13,40]
>BRU
BRU>/NOINITIALIZE/SUPERSEDE/NOPRESERVE/BACKUP_SET:MPLUSBL15SRC
FROM: MM0:[13,10]F11PRE.MAC,WTRN1.MAC
TO: DB0:
BRU--COMPLETED
BRU>^Z
```

Create the correction file:

```
>SET /UIC=[13,40]
>EDT WTRN1.COR
        .
        .
(Create correction file)
        .
        .
```

Apply the correction:

```
>SET /UIC=[13,10]
>SLP @[13,40]WTRN1.COR
```

Assemble the corrected module with the Executive macro library, the executive prefix file RSXMC.MAC, and the prefix file F11PRE.MAC:

```
>SET /UIC=[13,24]
>MAC WTRN1=[1,1]EXEMC/ML,[11,10]RSXMC,[13,10]F11PRE,WTRN1
```

Replace the defective module in the FCP library:

```
>SET /UIC=[1,24]
>LBR FCP/RP=[13,24]WTRN1
MODULES REPLACED
WTRN1
```

All of the task build command files require that the logical device MP: be assigned to the appropriate device. Depending on the size of the patch, it may be necessary to change the value in the partition specification contained in the task build command file:

```
PAR=FCPPAR:0:nnn
```

In the following examples, xxx must be replaced by the 3-character designation for your desired FCP, that is, MDL or LRG.

Task-build the new FCP using the updated library:

```
>ASN NL:=MP:
```

Edit the task-build command file, if necessary, at this time.

```
>TKB @FCPxxxBLD
```

Install the updated FCP in the system image:

```
>SET /SYSUIC
SYSUIC=[g,54]
>SET /UIC=[g,54]
>PIP /NV=[1,54]FCPxxx.TSK
>RUN $VMR
ENTER FILENAME: RSX11M
VMR>REM F11ACP
VMR>INS FCPxxx/PAR=GEN/IOP=NO/CKP=NO
VMR>^Z
>RUN $SHUTUP
```

Reboot the system to place the new FCP in use.

1.4.2.5 **Applying Corrections to Reconfiguration Tasks CON and HRC** - The following example illustrates how to patch module CNCMR for the CON task. Unless indicated otherwise, use a similar command sequence for correcting module HRONL for the HRC task.

This example assumes that:

- You have deleted the source files in [27,10].
- You have not deleted the object libraries and command files in [1,24] and [1,20].
- The object library for both CON and HRC is [1,24]OLR.OLB.
- The disk to which you will apply the patches is mounted Files-11 in drive DB0:.
- The distribution tape is mounted foreign on MM0:.

Perform the following steps to patch CNCMR:

1. Boot your system, and log in to a privileged account.
2. If necessary, restore the required files from the distribution tape.

```
>UFD DB0:[27,10]
>UFD DB0:[27,40]
>UFD DB0:[27,24]

>BRU
BRU>/NOINITIALIZE/SUPERSEDE/NOPRESERVE/BACKUPSET:MPLUSBL15SRC
From: MM0:[27,10]CNPRE.MAC,CNCMR.MAC
To:  DB0:
BRU -- Completed
```

   If you were correcting module HRONL, you would substitute for the From: line in the previous command sequence:

```
MM0:[27,10]HRPRE.MAC,HRONL.MAC
```

3. Create the SLP correction file in UIC [27,40].

```
>SET /UIC=[27,40]
>EDT CNCMR.COR
      .
      .
      .
```

4. Apply the SLP correction file to CNCMR:

```
>SET /UIC=[27,10]
>SLP @[27,40]CNCMR.COR
```

5. Assemble the corrected CNCMR module using the Executive macro library and the Executive prefix file RSXMC.MAC: you also use these when assembling a corrected HRC module. In addition, use either the prefix file CNPRE.MAC if assembling a CON module or HRPRE.MAC if assembling an HRC module.

```
>SET /UIC=[27,24]
>MAC CNCMR=[1,1]EXEMC/ML,[11,10]RSXMC/PA:1,[27,10]CNPRE,CNCMR
```

6. Replace the CNCMR object module in the OLR library:

```
>SET /UIC=[1,24]
>LBR OLR/RP/NOEP=[27,24]CNCMR.OBJ
```

7. Task-build CON using the updated library:

```
>ASN DB0:=IN:
>ASN DB0:=OU:
>ASN NL:=MP:

>TKB @CONBLD
```

NOTE

If you do not have the task-build command files [1,24]CONBLD.CMD and [1,24]CONBLD.ODL, you must use the nonprivileged task-build section of SYSGEN in order to rebuild CON. For an HRC module, you must rebuild HRC if you do not have [1,24]HRCBLD.CMD and [1,24]HRCBLD.ODL.

8. Install the updated reconfiguration utility in the system image:

For CON, type:

```
>SET /SYSUIC
SYSUIC=[g,54]
>SET /UIC=[g,54]
PIP /NV=[1,54]CON.TSK
RUN $VMR
Enter filename: RSX11M
VMR>REM ...CON
VMR>INS CON
VMR>^Z
```

For HRC, type:

```
>SET /SYSUIC
SYSUIC=[g,54]
>SET /UIC=[g,54]
PIP /NV=[1,54]HRC.TSK
RUN $VMR
Enter filename: RSX11M
VMR>REM HRC...
VMR>INS HRC
VMR>^Z
```

9. In order to use the new reconfiguration tasks,

   - Shut the system down using the shutup utility:

     ```
     >RUN $SHUTUP
     ```

   - Reboot the system.

1.4.2.6 **Applying FCS Corrections** - Correcting the FCS modules on an RSX-11M-PLUS system can be done by updating the source files,

assembling them, and replacing modules in the system library, usually LB:[1,1]SYSLIB.OLB. This process is complicated by the fact that there are three kinds of FCS:

1. ANSI - supports ANSI format magnetic tape and big buffers
2. Non-ANSI - does not support ANSI tape or big buffers
3. Multibuffered - supports ANSI tape, big buffers, and multiple buffers.

An FCS source file like CLOSE.MAC contains conditional assembly directives that can produce three different CLOSE objects, depending on the global symbols defined when CLOSE.MAC is assembled. (These three different CLOSE objects correspond to the three kinds of FCS.) Other FCS source files, like DELETE.MAC, have no such conditional assembly directives. They are only assembled one way; that is, only one DELETE object exists.

The SYSLIB.OLB provided on the kits contains the ANSI FCS. Thus, this SYSLIB contains modules like CLOSE, assembled with the ANSI tape conditionals and big buffer conditionals enabled, and modules like DELETE, which have no such conditionals and are the same in any FCS.

An alternate system library called NOANSLIB.OLB is also provided. It contains an FCS that does not support ANSI tape, big buffers, or multiple buffers. Be sure that you know whether the SYSLIB on your system contains the ANSI FCS, or if it has been replaced with the non-ANSI or multibuffered FCS.

Details of the correcting procedure follow. MCR syntax is used throughout.

Updating the FCS Sources - The FCS source files are found in [50,10] on the kits. Updating a source is done by entering the correction file into [50,10], and entering "SLP @filename" to apply the correction.

Assembling FCS - Assembling the updated source(s) can be done in either of two ways. One way is simple and time-consuming; the other is quick, but it must be done with great care. The simple way is to set your UIC to [50,20] and enter MAC @FCSASM. This will assemble every FCS variant properly, producing over 100 object files.

The other choice is to assemble only the sources that have been updated. This is not as simple, but it saves machine time. To assemble only a particular file, look at the five *.CMD files in [50,20] on the kit. Search the command files for all mentions of the file you want to assemble. A description follows of each command file and what you do with its contents.

1. FCSBOTH.CMD assembles files like DELETE.MAC, which contain no code specifically written to support or deny support to ANSI tape, big buffers, or multiple buffers. If the source file you have updated is mentioned in FCSBOTH.CMD, then it is assembled the same way regardless of which kind of FCS is in your SYSLIB. For example, this is the command line that assembles DELETE:

   ```
   [50,20]DELETE,[50,30]DELETE/-SP=-
   [50,10]FCSPRE,DELETE
   ```

(FCSPRE.MAC defines necessary macros and global symbols.) To assemble DELETE.MAC, you should set your UIC to [50,20] and enter the following:

```
MAC [50,20]DELETE,[50,30]DELETE/-SP=[50,10]FCSPRE,DELETE
```

As an alternative, you could put the command line in a file and enter "MAC @filename". A file like DELETE.MAC, which has no conditional assembly directives, will be mentioned only in FCSBOTH.CMD.

2. FCSANSI.CMD assembles source files that contain ANSI tape, big buffer, or multiple-buffering conditionals. It produces objects for the ANSI kind of FCS. If your SYSLIB contains the ANSI FCS, and the source file you have updated is mentioned in FCSANSI.CMD, then you should use the command line you found in FCSANSI.CMD to assemble the source file. For example, this command line from FCSANSI.CMD assembles CLOSE.MAC:

   ```
   [50,20]CLOSE.MTA,[50,30]CLOSE.MTA/-SP=-
   [50,10]FCSANSI/PA:1,FCSBIGBUF/PA:1,FCSPRE,CLOSE
   ```

   FCSANSI.MAC enables the ANSI tape conditional assembly directives, and FCSBIGBUF.MAC enables the big buffer conditionals. Note the sequence of input file names. It is absolutely imperative that the source file is the last file name specified, and that FCSPRE is next to it, in every FCS assembly. Otherwise, the proper conditionals will not be enabled, with potentially confusing and dangerous results. Any file that is assembled by FCSANSI.CMD is also assembled by FCSNOANSI.CMD and FCSMULBUF.CMD. Note the "MTA" file types above, which distinguish between the three kinds of CLOSE objects.

3. FCSNOANSI.CMD also assembles source files that contain ANSI tape, big buffer, or multiple-buffering conditionals. It produces objects for the non-ANSI kind of FCS. If your SYSLIB contains the non-ANSI FCS, and the source file you have updated is mentioned in FCSNOANSI.CMD, then you should use the command line you found in FCSNOANSI.CMD to assemble the source file. This command line from FCSNOANSI.CMD assembles CLOSE.MAC:

   ```
   [50,20]CLOSE.NMT,[50,30]CLOSE.NMT/-SP=-
   [50,10]FCSPRE,CLOSE
   ```

4. FCSMULBUF.CMD assembles source files that contain ANSI tape, big buffer, or multiple-buffering conditionals. It produces objects that support all of these features. This command line from FCSMULBUF.CMD assembles CLOSE.MAC:

   ```
   [50,20]CLOSE.MBF,[50,30]CLOSE.MBF/-SP=-
   [50,10]FCSANSI/PA:1,FCSMULBUF/PA:1,FCSBIGBUF/PA:1,FCSPRE,CLOSE
   ```

   FCSMULBUF.MAC enables the multiple-buffering conditionals.

5. FCSASM.CMD simply causes the other command files to be executed, to assemble FCS in all ways. If you are assembling single sources, ignore FCSASM.CMD.

More information is necessary to correctly assemble a few FCS sources that require additional prefix files. These sources are GET.MAC, PUT.MAC, OPEN.MAC, FINIT.MAC, and RDWRIT.MAC. GET.MAC can be

assembled in the normal way (ANSI, non-ANSI, or multibuffered) to produce GET.MTA, GET.NMT, or GET.MBF. An additional file, GPSEQ.MAC, will enable conditional assembly directives in GET.MAC to produce GETSQ, the sequential GET module, for each kind of FCS. For example, this is the command line from FCSANSI.CMD, which produces GETSQ.MTA:

```
[50,20]GETSQ.MTA,[50,30]GETSQ.MTA/-SP=-
[50,10]GPSEQ/PA:1,FCSANSI/PA:1,FCSBIGBUF/PA:1,FCSPRE,GET
```

If you correct GET.MAC, be sure that you assemble it to produce objects GET and GETSQ, for whatever kind of FCS your SYSLIB contains.

Similarly, GPSEQ.MAC will produce PUTSQ, the sequential PUT module, for each kind of FCS. This is the command line from FCSNOANSI.CMD, which produces PUTSQ.NMT:

```
[50,20]PUTSQ.NMT,[50,30]PUTSQ.NMT/-SP=-
[50,10]GPSEQ/PA:1,FCSPRE,PUT
```

If you correct PUT.MAC, be sure that you assemble it to produce objects named PUT and PUTSQ.

OPEN.MAC can be assembled in the normal way (ANSI, non-ANSI, or multibuffered) to produce OPEN.MTA, OPEN.NMT, or OPEN.MBF. Three additional prefix files exist.

OPFID.MAC produces the open-by-file-id module, OPFID. This is the command line from FCSANSI.CMD, which produces OPFID.MTA:

```
[50,20]OPFID.MTA,[50,30]OPFID.MTA/-SP=-
[50,10]OPFID/PA:1,FCSANSI/PA:1,FCSBIGBUF/PA:1,FCSPRE,OPEN
```

OPFNB.MAC produces the open-by-file-name block module, OPFNB. This is the command line from FCSANSI.CMD, which produces OPFNB.NMT:

```
[50,20]OPFNB.NMT,[50,30]OPFNB.NMT/-SP=-
[50,10]OPFNB/PA:1,FCSPRE,OPEN
```

OPENR.MAC produces the open module for resident libraries, OPENR. This is the command line from FCSMULBUF.CMD, which produces OPENR.MBF:

```
[50,20]OPENR.MBF,[50,30]OPENR.MBF/-SP=-
[50,10]OPENR/PA:1,FCSANSI/PA:1,FCSMULBUF/PA:1,FCSBIGBUF/PA:1,FCSPRE,OPEN
```

If you correct OPEN.MAC, be sure that you assemble it to produce objects OPEN, OPFID, OPFNB, and OPENR, for whatever kind of FCS your SYSLIB contains.

FINIT.MAC and RDWRIT.MAC can be assembled to produce FINIT.OBJ and RDWRIT.OBJ. The prefix file FCSSUP.MAC enables conditionals to produce objects FINTSL.SUP and RDWRSL.SUP to be used in supervisor-mode libraries. These objects are necessary for correct execution of the user AST completion routines, which may be specified for FCS READ$ and WRITE$ functions. This is the line from FCSANSI.CMD, which produces FINTSL.SUP:

```
[50,20]FINTSL.SUP,[50,30]FINTSL.SUP/-SP=[50,10]FCSSUP/PA:1,FCSPRE,FINIT
```

Object files FINTSL.SUP and RDWRSL.SUP contain the same entry point names as FINIT.OBJ and RDWRIT.OBJ. Therefore they must be replaced in SYSLIB with their entry points deleted, so that they are only used when they are explicitly called when a supervisor-mode library is task-built. See the example below.

Replacing the FCS Object Modules - Set your UIC to [1,1], make a backup copy of the libraries, and use LBR to replace the corrected FCS objects that you have assembled. For example, suppose you have updated and assembled DELETE, PUT, and FINIT on a system with an ANSI FCS in SYSLIB:

```
SET /UIC=[1,1]
PIP /NV=SYSLIB.OLB
LBR SYSLIB/RP=[50,20]DELETE.OBJ,PUT.MTA,PUTSQ.MTA
LBR SYSLIB/RP=[50,20]FINIT.OBJ,FINTSL.SUP/-EP
```

Suppose you also use NOANSLIB.OLB:

```
PIP /NV=NOANSLIB.OLB
LBR NOANSLIB/RP=[50,20]DELETE.OBJ,PUT.NMT,PUTSQ.NMT
LBR NOANSLIB/RP=[50,20]FINIT.OBJ,FINTSL.SUP/-EP
```

If you wish to incorporate the corrected modules into an FCSRES resident library or FCSFSL supervisor-mode library, rebuild the library, and then rebuild every task that links to it with SYSGEN. If you rebuild and install a resident library and run a task that linked to the old resident library, then the task will call routines in the resident library at the wrong addresses. The results will be uncertain and potentially damaging, especially for privileged tasks.

1.4.3 Patching Object Modules

The Object Module Patch Utility (PAT) incorporates an object patch into an existing object module. DIGITAL publishes a patch file written in MACRO-11 assembly language to correct an error or make a change. You then create and assemble the published file and input it to PAT along with the object file being patched.

You must include the published checksum values in the specification for the input file and the correction files. If the checksum value does not agree with the computed result, PAT reports an error. If the correction file caused the error, check the source file against the original published source file to verify that it was copied correctly. If the input file caused the error, verify that the correct version of the file is being patched.

Note that it normally is necessary to use LBR to extract the original object module from a library. PAT produces a new object module file containing the corrected object code. When you patch modules from a library, take the object module from the distributed library, correct it, and then put it into a copy of the original library. (All corrections are put in the copy, which is used to rebuild the task.) The following guidelines apply to the formats for PAT command input and correction files.

Specify an explicit name, extension, and version number for each file. Use the following conventions for names:

File	Specification
Correction file source	name.PAT
Correction file object	name.POB
Input file object	name.OBJ;n
Corrected object file	name.OBJ;n+1

Specify checksums for all input and correction files.

The following example illustrates the command formats for creating the correction file and the PAT output file:

```
>MAC NAME.POB=NAME.PAT
>LBR NAME.OBJ;1=LIBRARY.OLB;1/EX:NAME
>PAT NAME.OBJ;2=NAME.OBJ;1/CS:xxxx,NAME.POB/CS:yyyy
>PIP LIBRARY.OLB;2=LIBRARY.OLB;1
>LBR LIBRARY.OLB;2/RP=NAME.OBJ;2
```

1.5 NOTES TO RSX-11M/M-PLUS DOCUMENTATION

The following sections contain notes that correct or supplement the RSX-11M-PLUS V2.1 documentation.

1.5.1 RSX-11M/M-PLUS MCR Operations Manual

The /TERM keyword for the MCR SET command now supports the LA50. You should add this new support to page 3-210 of Chapter 3.

1.5.2 RSX-11M/M-PLUS Utilities Manual

On page 3-21 of Chapter 3, the Peripheral Interchange Program (PIP) Utility, the maximum value of the byte location of EOF is 377(8). Any value over this results in an error message.

In Chapter 4, the File Transfer Utility (FLX), Table 4-1 page 5, the following correction should be made:

/IM:n	Image Mode	The transfer is to be in image mode. Image mode forces fixed-length records. You can use the value n to indicate the desired record length (in octal bytes) for Files-11 output (1000(8) bytes maximum). If you do not specify n, FLX assumes a record length of 512(10) bytes. The documentation currently indicates the value of n in decimal rather than in octal bytes.

On page 10-12 of Chapter 10, the Librarian Utility Program (LBR), the format for specifying the Create Switch (/CR) is as follows:

For macro and object libraries:

```
outfile/cr[:size:ept:mnt:infiletype=infile]
```

For universal libraries:

```
outfile/cr[:size:ept:mnt:infiletype:libtype=infile]
```

In Chapter 11, the File Dump Utility Program (DMP), the following new switch should be added to Table 11-1.

Switch	Description
/LIM:n:m	Specifies the range of bytes n through m of each record or block to be dumped. /OCT is still the default if no format switches are specified.

In Chapter 14, the Object Module Patch Utility (PAT), the following new error message should be added to PAT MESSAGES:

UNABLE TO OPEN FILE filename

Explanation: There is insufficient work space in the internal File Storage Region (FSR) of the PAT utility.

User Action: Install or Run the PAT utility with an increment.

1.5.3 IAS/RSX-11 I/O Operations Reference Manual

1.5.3.1 **MTAACP Enhancements** - The following sections summarize the enhancements made to the RSX-11M-PLUS V2.0 and V2.1 Magnetic Tape Ancillary Control Processor (MTAACP). Detailed descriptions of V2.0 enhancements are included in Appendix G of the IAS/RSX I/O Operations Reference Manual.

The RSX-11M-PLUS V2.0 MTAACP has been enhanced for the following:

- Compliance with the ANSI and FIP standards
- Processing of unlabeled tapes
- Mount, dismount, and volume switch control

The RSX-11M-PLUS V2.1 MTAACP has been enhanced to support RMS-11 V2.0.

Compliance with ANSI and FIPS - DIGITAL now provides tape-labeling systems that conform to ANSI Standard X3.27 -- 1978, according to the Federal Information Processing Standard. The MTAACP has been enhanced to comply fully at levels 1 and 2, and to comply at level 3 with the exception of support for user-supplied labels. User-supplied labels may appear on a tape; however, they are only accessible to application programs through the unlabeled tape facility.

The following new features are a result of the implementation of ANSI compliance:

- Seventeen-character file name support

  If file names appear in quotes ("), they may be up to 17 characters long and may contain any characters in the ANSI "a" character set.

  Note that, as a result of this change, PIP directory listings of magnetic tapes have a new format. The most noticeable difference is the presence of several spaces preceding the version number. The spaces are included to clearly delimit the version number from the file identifier, which, in accordance with the standard, may contain embedded semicolons.

- Volume set identification support

  Previously, the MTAACP only supported volume set identifiers that were the same as the volume identifier of the first volume in the set. However, the ANSI standard requires that volume identifiers be unique within an installation (thus preventing a user from choosing names of his or her own).

Full support of the volume set identifier allows the user to choose a name for this field, which becomes the primary means of access to a volume using the MOUNT command.

This change was implemented with full compatibility with the previous releases and requires no changes in procedures for users who do not wish to use the new feature.

- Control of attributes

  MOUNT and the MAG utility allow the specification of block and record sizes for tapes written on systems that do not write this information on the tape. They also allow control of the carriage control attribute for tapes created on systems whose system code is not recognized by the MTAACP.

  Note that the default block size for files written to tapes from an RSX system may also be modified with MOUNT or the MAG utility. This value is normally 512 bytes; if it is modified, additional buffer space must be allocated in the programs writing to tape.

- Magnetic tape file block size

  The PIP block size switch is no longer necessary when reading files from magtape; FCS will adapt automatically to input files of varying block size.

  PIP directory listing of files on magtape show the number of magtape blocks. Thus, the file size on a magnetic tape (represented in terms of variable size blocks) may be different from the size of the file on disk.

- Volume and file accessibility

  In accordance with the ANSI standard, volumes that have a nonspace character in the volume accessibility byte may be mounted only by a user who has the privilege to override accessibility.

  The Initialize Volume Utility allows the specification of the character to be placed in the volume accessibility byte in the Volume Header Label.

  Also in accordance with the standard, files that have a nonspace character in the file accessibility byte may only be accessed by a user who has mounted the volume with "override accessibility" specified. A user who has mounted a volume in this manner may also write a nonspace character in the file accessibility byte of the file header label by specifying a valid "a" character in the high byte of the file protection word and a low byte of all ones. The file protection word may be specified at MOUNT time (for example, a file protection specification of [,,WED,RWD] represents the character "A") or as an attribute to the CREATE QIO.

Support of Unlabeled Tapes - The MTAACP now permits tapes to be mounted with the /NOLABEL switch. This allows utilities such as PIP and higher-level language applications programs (in FORTRAN, for example) to read and write unlabeled tapes using the standard READ and WRITE statements in the language.

Files on unlabeled tapes may contain fixed-length records that may be blocked or unblocked.

In addition, when processing unlabeled tapes, you may request character set translation. An EBCDIC translate table is provided with the ACP; you may add up to three additional installation-defined translate tables. Control of this feature is provided by the MOUNT command and by the MAG utility, as well as by attributes specified by applications programs when you create files.

The file name used when you access an unlabeled tape may specify positioning information, if desired. If no positioning information is desired, the file name has no meaning. However, some utilities, such as PIP, may require a dummy file name to prevent the use of wildcard mode.

Note that there are now three ways tapes may be mounted:

- Mounted for ANSI
- Mounted for unlabeled processing
- Mounted foreign

Access to a tape mounted "foreign" is handled by logical QIO directives. Access to a tape mounted "unlabeled" (/NOLABEL) or "ANSI" is through virtual QIOs, primarily those issued by FCS, RMS, the READ and WRITE statements of higher-level languages, and file-oriented utilities such as PIP.

Mount, Dismount, and Volume Switch Control - When the operator dismounts an ANSI tape, the default action recommended in the standard is to unload the tape. Depending on which model tape drive is in use, this will either take the drive off line or actually unload the tape.

The DCL /[NO]UNLOAD switch and the MCR /LOCK=[NO]UNLOAD switch specify a different action to be taken. This switch can be applied to either the MOUNT or DISMOUNT command.

When a volume switch is requested, you may, by using the MAG utility, cause error codes to be returned to the program that is reading or writing. (Previously, it was necessary either to mount a tape or to abort the program.) The MAG utility also allows a tape to be initialized if a new tape is needed for output and no previously initialized tape is available.

RMS-11 Support - MTAACP supports new file attribute codes (creation, revision, expiration, and backup dates), which read and write the date in the file header using a 64-bit date format. MTAACP performs a computation to convert the date between ASCII and 64-bit binary form.

1.5.3.2 **FDB Offset Definitions** - The following information should be added to the FDB Offset definitions listed in Appendix A of the IAS/RSX-11 I/O Operations Reference Manual:

Offset	Size (in Bytes)	Contents
F.RATT	1	Bit 2 = 1 to indicate the "print file format" (FD.PRN). FCS allows this attribute but does not interpret the format word.
F.NRBD	4	Contains the next record buffer descriptor. The format of the record buffer descriptor is the size word followed by the address of the buffer.
F.RCNM	4	Contains the number of the record for random access operations. The format of the record number is the high-order word followed by the low-order word.
F.VBN	4	Contains the virtual block number. The format of the virtual block number is the high-order word followed by the low-order word.

1.5.3.3 **Shared Access and Block Locking** - Include the following information in Chapter 2, Section 2.2.1.5 of the IAS/RSX-11 I/O Operations Reference Manual:

facc

- FA.SHR - Shared access is also a precondition for block locking.

actl

- FA.LKL!FA.EXL - FCS permits limited block locking to coordinate two or more tasks' access to the same file. All tasks accessing the file must open the file for shared access by setting bit FA.SHR in FDB field F.FACC (the field access byte).

1.5.3.4 **Opening a File by File ID Number** - Include the following information in Chapter 3, Section 3.5 of the IAS/RSX-11 I/O Operations Reference Manual. This information follows the bulleted list at the end of the section:

The OFID$W macro call is equivalent to the OFID$U macro call; invoking either OFID$W or OFID$U opens an existing file by file ID number for update and extension.

1.5.3.5 **Buffer Flush Routine (.FLUSH)** - Include the following information in Chapter 4 of the IAS/RSX-11 I/O Operations Reference Manual as Section 4.17:

When using PUT$S to a disk or ANSI tape file, FCS does not normally write the contents of the block buffer to a file until the block buffer is full or the file is closed. The .FLUSH routine forces the block buffer to be written to the file. You may need this function to protect data from a system crash or if the data needs to be written so that another task can read it from the file.

The following register must be preset before calling this routine:

R0 must contain the address of the associated FDB.

During output, all registers are preserved, the carry bit is clear or set to indicate success or failure, and the FDB F.ERR byte contains the success or failure code.

1.5.3.6 **I/O Error Codes** - Updated I/O error codes for this manual are listed in the RSX-11M-PLUS Mini-Reference.

1.5.4 IAS/RSX-11 System Library Routines Reference Manual

The following program section names and SYSLIB routines should be added to Table 1-1 of the IAS/RSX-11 System Library Routines Reference Manual.

Program Section Name	SYSLIB Routines Module Name	Routine Name(s)
PUR$D	CAT5B (data)	$CAT5B
	EDTMG (data)	$EDTMG
PUR$I	CAT5B (instruction)	$CAT5B
	EDTMG (instruction)	$EDTMG

1.5.5 RSX-11M/M-PLUS ODT Reference Manual

The format for the relocation registers in this manual is incorrect. The correct format for the registers is $nx, where n is a number and x is a letter. The manual, however, incorrectly reverses the letter and the number, using an $xn format. For example, where the text references relocation register $R0, the reference should be $0R.

1.5.6 RSX-11M/M-PLUS Executive Reference Manual

The following changes should be made to the RSX-11M/M-PLUS Executive Reference Manual. All the changes apply to Chapter 5.

- On page 5-7, the first sentence under "Macro Expansion:" should read "Most of the directive descriptions expand the $ form of the macro."

- ALTP$

  On page 5-11, add the following return code and definition to the list of DSW return codes:

  IE.RSU -- Resource (the task's header) unavailable because task is checkpointed with outstanding I/O.

- ATRG$ and CRRG$

  On pages 5-17 and 5-44, the directive error status IE.PNS is returned by the Executive when the region specified in the directive is a main partition. You should add this information to the existing explanation of IE.PNS.

- CINT$

  1. On page 5-18, add the following text to the last paragraph:

     > However, be aware that including the symbol definition file can cause references to system subroutines to be resolved from that file instead of from the system library. To avoid this problem, explicitly include the required library modules before specifying the Executive symbol definition file. Specify the /SS switch with the file so the Task Builder will resolve any symbols that are still undefined. (Specifying the /SS switch prevents the Task Builder from trying to use multiply defined symbols.)

  2. For corrections to the CINT$ example, beginning on pages 5-24, see the June 1980 issue of the Software Dispatch, Sequence 7.1.1.1 N, pages 79 through 84.

- CPCR$

  On page 5-34, the local symbol C.PCR should be C.PCNM.

- GCCI$

  On page 5-76, the Command Buffer Format list should be in the following order:

```
G.CCDV -- . . .
G.CCUN -- Octal unit number of issuing terminal (1)
G.CCCT -- Number of characters (1)
G.CCCL -- . . .
G.CCTC -- Terminator (1)
G.CCFL -- Flags (1)
            .
            .
            .
G.CCBF -- . . .
```

- GCII$

  On page 5-80, add the following Local Symbol Definitions:

```
G.CIBF -- Buffer address
G.CIBL -- Buffer length
G.CICN -- Radix-50 CLI name
G.CIDV -- ASCII terminal name
G.CIUN -- Terminal unit number
```

- MSDS$

  On page 5-108, the Macro Expansion should be

```
MSDS$           mask
.BYTE           201.,2  ;DIC=201. DPB SIZE=  2 WORDS
.WORD           MASK
```

- QIOW$

  On page 5-116, the parameter efn (event flag number) is mandatory in the FORTRAN call.

- RDAF$

  On page 5-125, the Macro Call should define buf as the address of a 4-word buffer.

- RPOI$

  1. On pages 5-130 and 5-131, the second paragraph in the definitions of itask and task should be changed to the following:

     On RSX-11M-PLUS systems, any task may specify a new name for the requested task as long as the requested task is not a CLI task.

     For both systems, the requested task (specified in the tname parameter) must be installed in the ...tsk format.

  2. On page 5-131, in the Macro Expansion, the line

     ```
     .ASCII  /dev/   ;ASCII NAME OF TI: OF REQUESTED TASK
     ```

     should be the last line in the expansion.

- RUN$

  On pages 5-140 and 5-141, the smg parameter in the FORTRAN Call and the Macro Call is mandatory.

- SCAL$S

  On page 5-146, the parameter err (address of error routine) should be added to the Macro Call as an optional parameter (SCAL$S saddr,caddr[,err]).

- SCLI$

  On page 5-148, in the Macro Expansion, the line

  ```
  .RAD50  /cli/   ;CLI NAME
  ```

  should be the last line in the expansion.

- SDAT$

  On page 5-150, the text of the NOTE should be replaced with the following text:

  When a local event flag is specified, the flag is set for the sending task. When a common event flag is specified, the flag is set for all tasks. When a group-global event flag is specified, the flag is set for all tasks within the specified group. For all event flags, a significant event is always declared.

- SDRC$

  On page 5-153, in the Macro Expansion, change the line

  ```
  .BYTE   2       ;EVENT FLAG NUMBER = 2
  ```

  to

  ```
  .WORD   2       . . .
  ```

  and delete the line

  ```
  .BYTE   16.     ;EXIT STATUS BLOCK CONSTANT
  ```

- SMSG$

  1. On page 5-163, in the Macro Expansion, change the following lines:

     ```
     Change SMSG$   SM.ERR  . . .  to SMSG$ SM.SER . . .
     Change .BYTE   DIC,8.  . . .  to .BYTE 171,8. . . .
     Change .WORD   SM.ERR  . . .  to .WORD SM.SER . . .
     ```

     and delete from the list of Local Symbol Definitions the line

     ```
     S.MERR -- Error Log Target Identifier
     ```

  2. On page 5-164, change the definition of IE.ULN to "Specified LUN is not assigned to a mass storage device."

- SNXC$

  On page 5-165, add the following Local Symbol Definitions:

  ```
  S.NXDV -- Device name
  S.NXUN -- Device unit number
  ```

- STIM$

  On page 5-196, in the list of offsets, S.TICS should be S.TISC.

- VRCD$

  On page 5-210, in the third paragraph, replace the sentence

  > For this reason, the storage you allocate within the buffer should be two words greater than the size of the data portion of the message specified in the directive.

  with

  > The two words are added to the buffer size you specify.

- VSDA$

  On page 5-216, add to the explanation of the Macro Call the parameter

  ```
  efn = Event flag number
  ```

The Local Symbol Definitions list should be in the following order:

```
S.DATN -- . . .
S.DABA -- . . .
S.DABL -- Buffer length (2)
S.DAEF -- Event flag number (2)
S.DASP -- Reserved (2)
S.DATI -- Reserved (2)
```

- VSRC$

  On page 5-220, add to the DSW Return Codes the error

  IE.IBS -- Invalid buffer size specified (greater than 255.)

1.5.7 RSX-11M/M-PLUS I/O Drivers Reference Manual

A complete RD51 Fixed Disk/RX50 Flexible Disk Subsystem consists of the fixed and flexible disks and an RQDX1 controller (RQDX1/RD51, RX50). This information should be included on page 5-4.1 of this manual.

Updated I/O error codes for this manual are listed in the RSX-11M-PLUS Mini-Reference.

1.5.8 RSX-11M/M-PLUS System Management Guide

The following error messages for the Virtual Monitor Routine (VMR) currently are not documented in the manual:

- No ICB pool space for CPU <cpu>

  The ICB pool space allocated for the CPU <cpu> has been exhausted.

- Bus switch not loaded

  You attempted to load a device requiring a bus run before you loaded the bus switch.

CHAPTER 2

RMS–11 VERSION 2.0 RELEASE NOTES

CHAPTER 2

RMS-11 VERSION 2.0 RELEASE NOTES

2.1 NEW FEATURES FOR RMS-11 VERSION 2.0

The following new features appear in the RMS-11 V2.0 access methods.

2.1.1 Support for Clustering and Task Independence

The memory-resident library RMSRES can be clustered with any other resident library that supports clustering; this capability allows RMS-11 to share task virtual memory with other libraries. See your FMS-11 and language documentation to determine whether a particular library supports clustering.

Task-independence means that once a program is linked with RMSRES, the library can be rebuilt or replaced without requiring that the task linked to it be rebuilt.

Linking with RMSRES requires use of the RMSRLX.ODL file. Include the element RMSROT in the .ROOT statement of your ODL file, and the indirect reference @LB:[1,1]RMSRLX in your ODL file. If you used RMSRLX.ODL in this manner with V1.8, no change should be required (if you made reference to the RMSALL factor, this is no longer necessary but will do no harm).

To use this library as a cluster library, you simply specify the CLSTR option when linking the task.

2.1.2 Segmented Resident Library

RMSRES has been partitioned into seven separate library segments, not all of which need to be physically resident in memory.

This partitioning allows "demand-paging" of the library segments: only those segments containing code that is, or has most recently been, in use must be resident; the remainder of the library segments are eligible for replacement by tasks that can use the physical memory.

Each of the seven segments is 8KB or less in size; not more than two segments are ever required simultaneously by a given user task. This feature reduces the minimum amount of physical memory required to run RMS-11 to about 16KB. If limited to this space, however, a task that mixes operations requiring support by multiple segments will generate quite a bit of disk activity as the segments "swap" against each other.

The RMS-11 segmented libraries are the following:

- RMSRES.TSK is the "root" of the segmented library. As before, this TSK file and its associated STB file are the only files needed to link user tasks against RMS-11.

- RMSLBA.TSK is the library segment that supports DISPLAY, ERASE, PARSE, SEARCH, RENAME, sequential file CREATE and OPEN, common portions of relative and indexed file CREATE and OPEN, and common CLOSE (and file operation exit) processing.

- RMSLBB.TSK is the library segment that supports sequential file CLOSE and all sequential file record operations, FREE, magnetic tape functions, block I/O record operations, and the common internal EXTEND function.

- RMSLBC.TSK is the library segment that supports all relative file operations, plus the EXTEND function.

- RMSLBD.TSK is the library segment that supports indexed file OPEN, CLOSE, CONNECT, DISCONNECT, FIND, and GET operations.

- RMSLBE.TSK is the library segment that supports indexed file DELETE/PUT/UPDATE operations.

- RMSLBF.TSK is the library segment that supports data record insertion and index-update for indexed PUT/UPDATE operations, plus the bucket allocation routine, and the indexed CREATE operation.

Note that although the library segments need not be physically resident, **all** library segments must be installed before the resident library can be used (and before the RMS-11 utilities, which are built against the resident library, can be used).

2.1.3 Supervisor-Mode Support

The version of RMSRES that is provided can also be used in supervisor mode on systems that include hardware support for supervisor mode.

To use supervisor mode, do the following:

- Include the modules R0EXSY, R0AUTS, R0IMPA, and RMSSYM from LB:[1,1]RMSLIB in your task root.

- Include the command RESSUP=LB:[3,54]RMSRES/SV:0 in your task-build command file. You cannot use SUPLIB since the resident library task image is located in LB:[3,54].

Note that the only difference between a task linked against RMSRES in user mode and a task linked against RMSRES in supervisor mode is that the RMSRES.TSK and RMSRES.STB files that are used for **linking** the task are different files. The actual RMSRES.TSK file that is installed in the system can be used in either mode.

2.1.4 Disk-Overlaid RMS-11

New versions of the RMS-11 V1.8 ODL files are provided on the distribution kit. These ODL files are:

- RMS11S.ODL

  Designed to use minimal virtual address space for support of sequential and relative file organizations, this ODL file adds about 6.5KB to the task size.

- RMS11X.ODL

  Designed to use minimal virtual address space for full support of all file organizations, this ODL file is structured to add about 10KB to the task size.

- RMS12X.ODL

  This ODL file adds about 12KB to the task size, supports all file organizations, and offers a good compromise between performance for record operations on indexed files and use of task virtual address space.

- RMS11.ODL.

  This is a prototype ODL file that you can tailor to your own needs.

The V1.8 ODL files will still work with RMS-11 V2.0 but the new versions of the ODL files will be more efficient, generally requiring less memory and less disk space for the task image. For more information on V1.8 comparisons, see Section 2.3.

In addition, two new ODL files are provided with V2.0:

- RMS12S.ODL

  Supporting only sequential and relative file organizations, this ODL file adds about 9KB to the task size and is designed to offer a good compromise between performance and use of task virtual address space.

- DAP11X.ODL

  Structured to add about 14KB to the task size, this ODL file is designed to use minimal virtual address space for support of all file organizations and (on systems with the required DECnet support) remote access facilities.

2.1.5 Remote File and Record Access via DECnet

If suitable DECnet facilities exist on your system and on the target system, RMS-11 V2.0 will allow file and record access to files residing on other network nodes, provided that such nodes include an RMS-11-based file access listener (FAL).

For most purposes, remote access is indistinguishable from local access, although performance may not be equivalent. The following general limitations apply:

- RMS-11 generally does not support remote functions that are not supported locally.

- Certain RMS-11 functions (wildcard support, PARSE, SEARCH, ENTER, REMOVE, RENAME, and transmission of device, directory, and file identifiers) are not supported by the data access protocol (DAP) used, and hence cannot be executed remotely.

- Certain FALs do not support the full set of RMS-11 functions expressible via DAP, and hence further limit remote access. For example, the current RSTS/E RMS-11 FAL does not support record access to indexed files.

- High-level languages may not allow expression of the file specification necessary to establish contact with a remote node.

To operate upon a remote file, your program must include a node specification for the remote file and you must include the RMS-11 remote access code when you build your task.

2.1.5.1 Remote Node Specification - You must include a remote node specification at the start of the file name string or the default name string you provide to an OPEN, CREATE, or ERASE operation. In addition, your file name string and default name string must conform to the DIGITAL Command Language (DCL) file specification syntax rules, and the file specification resulting from the merging of these two strings must conform to the file specification conventions on the destination system as well.

In general, a file specification consists of the following elements (in the order and with the delimiters shown):

```
node::device:[directory]filename.type;version
```

where the elements beyond the **node** specification conform to the conventions of the target system. Elements not present will be defaulted according to the conventions of the target system.

The **node** element is of the form:

```
node"user password"::
```

where **node** is the (required) destination node name and **"user password"** is an (optional) access control string containing login information acceptable to the remote system.

If **"user password"** is provided, the device and directory defaults and access privileges of that remote account are acquired; if not, the device and directory defaults and access privileges of the default DECnet account on the remote system are acquired.

2.1.5.2 Linking Programs with RMSDAP Support - To link your program with remote RMSDAP modules, you can either use the disk-overlaid version or reference the DAPRES resident library:

- To link with the disk-overlaid version, reference the DAP11X.ODL file in your ODL file. As with other RMS-11 ODL files, you reference the RMS-11 root modules as RMSROT, and specify other RMS-11 modules as the cotree RMSALL.

- To link with the DAPRES resident library, use the ODL file DAPRLX, and specify RMSRES and DAPRES in your TKB command file. RMSRES and DAPRES can be specified as single resident libraries or as clustered resident libraries.

Note that DAPRES is a clusterable, task-independent library as defined previously for RMSRES. It is not a segmented library, however.

2.1.5.3 RMSDAP Environments - RMS-11-based FALs are currently available on VAX/VMS, RSTS/E, RSX-11M, and RSX-11M-PLUS systems. The version of data access protocol that is used must be Version 5.6 or later; this means that you must have at least Version 2.0 of DECnet/E, Version 3.1 of DECnet for RSX-11M, Version 1.1 of DECnet for RSX-11M-PLUS, or Version 2.0 of DECnet/VAX.

For more information on remote access using RMS-11, see Appendix B of the RSX-11M/M-PLUS RMS-11 User's Guide.

2.1.6 New File and Directory Operations

RMS-11 V2.0 supports several new file and directory operations.

2.1.6.1 ENTER/REMOVE - Your program can use the ENTER operation to create a directory entry for a file. You can delete the directory entry for a file by using the REMOVE operation; this operation does not affect either the existence of the file or the file contents.

2.1.6.2 RENAME - RENAME is a new RMS-11 operation that can be used to replace the directory entry for a file (that is, change the name of a file). The file specification for the new directory entry must not specify a new device or node for the file, but can specify a different directory, file name, file type, and file version.

2.1.6.3 PARSE - PARSE is a new RMS-11 operation that can be used to analyze a file specification.

The PARSE operation merges the file specification with the default file specification and with the system defaults, and analyzes the result. If you specify wildcarding, the PARSE operation constructs a match-pattern; this match-pattern can be used by later wildcard SEARCH operations.

2.1.6.4 SEARCH - SEARCH is a new RMS-11 operation that can be used to find the next (or first) file that matches a match-pattern (constructed as the expanded string by a previous PARSE operation).

2.1.7 Wildcarding

You specify wildcarding by including wildcard characters in the file specification (or the default file specification). Wildcards can be specified in the directory entry, file name, type, and version.

See the RSX-11M/M-PLUS RMS-11 Macro Programmer's Guide for a full explanation of how to use the new directory operations and wildcarding. You may also want to examine the demonstration programs that are provided.

2.1.8 Random Access to Fixed-Format Disk Sequential Files

For disk sequential files with records of fixed format, random access for GET, FIND, and PUT operations can be specified. This kind of random access is very similar to random access to relative files, with some exceptions.

Your program specifies key access by setting the RB$KEY code in the 1-byte RAC field of the RAB, and by specifying the address of a "relative record number" in the KBF field of the RAB. Key size must be specified as 4 in the KSZ field of the RAB.

Some differences from relative file handling are the following:

- There is no initialization of the space in the file. If the first operation is a random PUT to the file at relative record number 5, the "cells" preceding record number 5 may contain "garbage": that is, whatever data happened to be there.

  (This is similar to VAX-11 RMS.)

- There is also no maximum record number (except as limited by the 4-byte specification of the key). If your program specifies a random PUT operation at relative record number 999, the file will be extended to the appropriate size (but not initialized).

- A PUT operation overwrites any existing record in the target location.

- There is no support for DELETE operations to a sequential file.

2.1.9 Increased Device Independence

Files with stream and VFC record formats can now be created on unit-record devices to avoid the need for special-case code in copy-type operations.

- In the case of VFC files, the record header is thrown away on output unless it is a "print format" file, which RMS-11 now supports as VAX-11 RMS does.

- For stream files, if none of the 3 carriage control bits is set (print format, carriage control, or FORTRAN carriage control), and if the last character is not a line feed, form feed, or vertical tab, then carriage-return/line feed (CR/LF) is appended at the end of the record.

- For stream files, where either the carriage control or FORTRAN carriage control attribute is set, if the last two characters of the record are CR/LF, the trailing CR/LF is stripped off, and then definition of the carriage control (CR or FTN) attribute is applied.

For similar ease-of-copying reasons, RMS-11 now allows creation of relative and indexed files for output to nondisk devices (although for magnetic tape, record format must be variable length or fixed length).

2.1.10 More Usable Block I/O Facilities

The following enhancements have been made for block I/O:

- Block I/O can now be used to copy files without the need to "fix up" the file header attributes.

  Block I/O READ operations now generally respect the logical EOF header attribute (exceptions are documented in the RSX-11M/M-PLUS RMS-11 User's Guide and in the RSX-11M/M-PLUS RMS-11 Macro Programmer's Guide).

- Block I/O can also be used for sequential access, by zeroing the RAB BKT field, allowing sequential copy and update operations.

- Extends are now performed automatically, when necessary, during block I/O access. The only exception is for sequential files that are accessed write-shared. In this case, automatic extends are not allowed because there is no mechanism for coordinating the updating of the file header attributes.

2.1.11 Sharing for Relative and Indexed Files

For increased symmetry, flexibility, and compatibility with VAX-11 RMS, RMS-11 V2.0 incorporates a change to previous file sharing semantics for relative and indexed files on RSX-11M/M-PLUS systems. This change affects only write-accessors who specify read sharing (the default) for record access to a relative or indexed file. Although it is largely transparent in functional terms, in some cases it will have a noticeable effect upon the performance of the writer.

In previous versions of RMS-11, the performance of the writer was dependent upon whether any write-sharing read-only accessors were already present when the writer requested access:

1. If such a reader was present, the writer obtained shared access to the file, although additional writers were prohibited. RMS-11 internal algorithms did not allow the writer to "cache" buckets unless the bucket remained locked; this limitation affected performance for certain types of operations.

2. If no such reader was present, the writer obtained exclusive access to the file. No subsequent RMS-11 readers could obtain access, regardless of their sharing specifications. RMS-11 internal algorithms allowed this writer to cache buckets when possible, thus improving performance by reducing disk I/O.

RMS-11 V2.0 has been changed to eliminate this access-order dependency:

1. When a write-accessor specifies read-sharing (the default), shared access is always granted as described in (1) above. Subsequent write-sharing readers can obtain access even if the writer was the first to access the file.

2. A writer desiring the performance advantages of exclusive access can specify "no sharing" (new for RMS-11 V2.0), in which case access is granted as described in (2) above.

The performance difference between case 1 and case 2 will be most evident for sequential PUT operations to a relative file with multiple

records in each bucket, or for sequential GET/UPDATE sequences to a relative file with multiple records per bucket or to an indexed file by primary key. If extra RMS-11 internal buffers are provided to improve performance, they will be ineffective for a writer unless "no sharing" is specified.

2.1.12 RMS-11 Utilities

2.1.12.1 New File Design Utility - The RMS-11 File Design Utility (RMSDES) is a new utility that allows you to interactively design and create files. RMSDES supersedes the RMS-11 File Definition Utility (RMSDEF).

Some of the features of RMSDES are as follows:

- Full RMS-11 file structure capabilities -- You can design and create any type of RMS-11 file to contain your data records, tailoring it to meet your program's data processing requirements exactly.
- Error detection -- RMSDES can detect many kinds of errors and omissions in the attribute information you supply. Because RMSDES is interactive, you can correct the errors immediately, before you create and load records into the file.
- Saved file description -- You can save a description of a file design for future use, whether or not you have created a file based on the description. This is useful if you want to complete an unfinished file design at a later date, or modify a file design to accommodate a new use.
- Copied file description -- RMSDES can construct a file description based on the attributes of an existing data file. You can either create a new file like the existing file, or modify the file design before creating a new file.
- Default calculations -- RMSDES can calculate defaults for many attributes.
- Extensive help facilities are provided for RMSDES.

2.1.12.2 New Features for RMSDSP - RMSDSP has a completely reorganized and much more readable format. It also includes the following new features:

- The combination of options /BP/FU now displays more complete information about the contents of backup container files.
- The size field lists high block as well.
- The protection information is "decomposed" into system, owner, group, and world protection fields.
- RMSDSP lists the file owner.

2.1.12.3 New Features for RMSCNV - RMSCNV has been rewritten and is smaller and generally faster than the 1.8 version. It has the following new features:

- The /CA switch can be used to dynamically create an output file, by copying file attributes from another file. (That is, you do not need to run RMSDES or RMSDEF to create the output file before doing the conversion if you already have a prototype file whose attributes can be copied for the output file.)

- The /IM switch can be used to copy files in image (block) mode.

- The /EO switch can be used to change ASCII stream files that end with a CTRL/Z character to end with null characters. This switch is provided for help in converting certain non-RMS-11 stream files into a format readable by RMS-11 V2.0. (See also Section 2.3.6, Other RMS-11 Version 1.8 Compatibility Notes.)

- The /KN:keyname switch can be used to convert files based on a key name (as opposed to a key number, as in the /KR switch).

- The /ML switch allows you to explicitly set the limit of memory allocation.

- The /SL switch, summary listing, has been significantly enhanced. It now produces several summaries: summaries of options used for the conversion, summaries of the file attributes for the files involved in the conversion, a message summary of errors and warnings, and a processing summary.

- Remote access capability is provided for suitable configurations. Your system and the target system must include DECnet support (at least V2.0 of DECnet/E). The target system must also include an RMS-11-based file access listener (FAL), and you must rebuild RMSCNV to include the remote access capability (see Section 2.8.2, Utility Configurations).

2.1.12.4 Additional Utility Features -

- RMSIFL has been rewritten and incorporates a number of bug fixes.

  In addition, the algorithms for allocating work file space for sorting have been changed; in most cases, the amount of space needed for the sort should be decreased.

  The /DE switch has been enhanced to allow specification of a different device for RMSIFL's temporary files.

- Command and ODL files are provided for rebuilding the RMS-11 utilities to use different forms of the RMS-11 access methods. See Section 2.8.2, Utility Configurations.

- All utilities now use lowercase format for messages and displays.

- The error message format has been changed for all utilities except RMSDEF and RMSIFL.

2.2 SUMMARIES OF TECHNICAL CHANGES

The following sections summarize the new software features for RMS-11. The summaries of technical changes also appear in the front matter of each manual.

The title of each manual is followed by a letter indicating whether the manual is a new (N) or revised (R) version.

In addition to the new and revised manuals listed in the following sections, the RMS-11 V2.0 documentation set includes a new manual, RSX-11M/M-PLUS RMS-11: An Introduction, which introduces RMS-11 concepts, terminology, and operations. The RSX-11M-PLUS Mini-Reference also contains a new section on RMS-11 utilities and error codes.

NOTE

All new RMS-11 features are fully accessible only to MACRO-11 programmers. See your high-level language documentation for supported features.

2.2.1 RSX-11M/M-PLUS RMS-11 User's Guide (R)

This revision of the RSX-11M/M-PLUS RMS-11 User's Guide documents the following technical changes:

- RMS-11 Version 2.0 supports random access to fixed-format disk sequential files and sequential block access to disk files of any format and organization.

- The RMS-11 Version 2.0 resident libraries are task independent. This means that once a program is linked with this library, the library can be rebuilt or replaced without requiring that the task linked to it be rebuilt.

- RMS-11 Version 2.0 contains no library equivalent to the RMSSEQ memory-resident library included with RMS-11 Version 1.8. The RMSRES resident library or the disk-resident ODL files can be used to obtain equivalent functionality and performance.

- New versions of the RMS-11 Version 1.8 ODL files are provided. These ODL files are RMS11S.ODL, RMS11X.ODL, RMS12X.ODL, and RMS11.ODL. The Version 1.8 ODL files will still work with Version 2.0, but the new versions will be more efficient. RMS-11 V1.8 ODL structures other than RMS11S.ODL, RMS11X.ODL, and RMS12X.ODL may not work correctly with the RMS-11 V2.0 code; when in doubt, verify them by comparison with the V2.0 RMS11.ODL file. In addition, two new ODL files are provided with Version 2.0: RMS12S.ODL and DAP11X.ODL.

- Files with stream and VFC records can now be created on unit-record devices to avoid the need for special-case code in copy-type operations.

  - For VFC files, the record header is thrown away on output unless the file is a "print format" file.

  - For stream files, if none of the 3 carriage control bits is set (print format, carriage control, or FORTRAN carriage control), and if the last character is not a linefeed, formfeed, or vertical tab, the carriage-return/linefeed (CR/LF) is appended at the end of the record.

  - For stream files, if either the carriage control or FORTRAN carriage control attribute is set, and if the last 2 characters of the record are CR/LF, the trailing CR/LF is stripped off and then definition of the carriage control attribute (CR or FTN) is applied.

- For similar ease-of-copying reasons, RMS-11 now allows creation of relative and indexed files for output to nondisk devices (for magnetic tape, however, the record format must be variable length or fixed length).

- The RMS-11 File Design Utility (RMSDES) is a new utility that allows you to design and create files interactively. It is fully documented in the RSX-11M/M-PLUS RMS-11 Utilities manual.

- RMS-11 Version 2.0 supports five new directory operations: ENTER, PARSE, REMOVE, RENAME, and SEARCH. These operations are fully documented in the RSX-11M/M-PLUS RMS-11 Macro Programmer's Guide.

- RMS-11 Version 2.0 supports a new wildcard file specification facility and a new print-record output handling format. These are also fully documented in the macro programmer's guide.

- User-provided interlocks allow a special, limited form of sequential file sharing among a group of accessors that includes at most one read/write accessor and any number of read-only accessors.

- If suitable DECnet facilities exist on your system and on the target system, RMS-11 Version 2.0 will allow file and record access to files on remote network nodes, if those nodes include an RMS-11-based file access listener (FAL).

- For magnetic tape, RMS-11 now allows fixed-format records to be less than 18 bytes.

- Files with stream or VFC records can now be created on unit-record devices. In addition, RMS-11 now allows the creation of relative and indexed files for output to nondisk devices, although they will be treated as sequential files.

- <CTRL/Z> and <ESC> are no longer recognized as record terminators for stream files, and <CTRL/Z> is no longer recognized as a file terminator for stream files.

- RMS-11 Version 2.0 pads stream files with null characters, to the high block of the file (not just to the end of the current block).

- The memory-resident library RMSRES can be clustered with any other resident library that supports clustering.

- On systems that include hardware support for supervisor mode, RMSRES can also be used in supervisor mode.

2.2.2 RSX-11M/M-PLUS RMS-11 Macro Programmer's Guide (R)

This revision of the RSX-11M/M-PLUS RMS-11 Macro Programmer's Guide documents the following technical changes:

- The new operation macros $ENTER, $PARSE, $REMOVE, $RENAME, and $SEARCH are documented, along with the related NAM block fields FNB, RSA, RSL, and RSS.
- The new facility for wildcard file specification is documented.
- The extension of access sharing is documented, along with the related masks FB$UPI and FB$NIL for the SHR field of the FAB.
- Random access to a sequential file with fixed-length records (similar to random access to a relative file) is documented.
- The new print-format record-output handling is documented, along with the related symbol FB$PRN for the RAT field of the FAB.
- The new sequential block access is documented; the previous block access (formerly called block I/O) is now called VBN access (virtual block number access).
- Block access can now be used to copy RMS-11 files without the need to modify the file's attributes manually.
- The addition of the success handler facility for file operation macros ($CLOSE, $CREATE, $DISPLAY, $ERASE, $EXTEND, and $OPEN) is documented.
- Increased device transparency for record access copy operations is supported. VFC and stream record formats are supported on unit-record devices. Relative and indexed files can be created for record access on nondisk devices, although they will appear as and be processed as sequential files there.
- The obsolete RMS-11 initialization macros $INIT and $INITIF are no longer documented. These macros are now defined as no-ops in the RMS-11 macro library RMSMAC.MLB; their previous functions are no longer needed because RMS-11 is now self-initializing. However, programs that use the $INIT and $INITIF macros in their previous senses remain valid under RMS-11 Version 2.0.
- Each XAB type now has a distinct name; the following are the new names:

ALL block	Area allocation XAB
DAT block	File date XAB
KEY block	File key XAB
PRO block	File protection XAB
SUM block	File summary XAB

- The following symbol declaration macros are documented:

FAB$BT	Declare FAB value and mask symbols
NAM$BT	Declare NAM block value and mask symbols
RAB$BT	Declare RAB value and mask symbols
XAB$BT	Declare XAB value and mask symbols
XBAOF$	Declare ALL block symbols
XBDOF$	Declare DAT block symbols
XBKOF$	Declare KEY block symbols

XBPOF$	Declare PRO block symbols
XBSOF$	Declare SUM block symbols

- The description of each operation macro includes the use and meaning of each associated control block field.
- The value of each RMS-11 user symbol is documented.
- The structure of each RMS-11 user control block is documented.
- The FAB has a new LRL (longest record length) field for sequential files (corresponding to the VAX-11 RMS XAB LRL field).
- The date/time XABs have changed in size from 36 to 46 octal bytes.
- For ANSI magnetic tape, RMS-11 allows fixed-format records to be less than 18 bytes.
- <CTRL/Z> and <ESC> are no longer recognized as record terminators for stream files; and <CTRL/Z> is no longer recognized as a file terminator for stream files.
- RMS-11 now pads stream files with null characters to the high block of the file (not just to the end of the current block). This means that RMS-11-created stream files can be read by programs that do not recognize the EOF value from the file header.

2.2.3 RSX-11M/M-PLUS RMS-11 Utilities (N)

The RSX-11M/M-PLUS RMS-11 Utilities manual is a new manual for this release, replacing Chapter 9 of the RMS-11 User's Guide for RMS-11 Version 1.8. It documents the following technical changes:

- The RMS-11 File Design Utility (RMSDES) is a new utility that allows you to interactively design and create files. RMSDES supersedes the RMS-11 File Definition Utility (RMSDEF).
- The RMS-11 File Display Utility (RMSDSP) provides more information about magnetic tape files and about backup files on disk and on magnetic tape (container files), and provides an optional detailed display for backup files. The displays also appear in a more readable format, and the size field shows both the allocated and used sizes of the file.
- The /DE switch to the RMS-11 Indexed File Load Utility (RMSIFL) has been enhanced to allow specification of a different device for RMSIFL's temporary files.
- The /SL switch to the RMS-11 File Conversion Utility (RMSCNV), the File Back-Up Utility (RMSBCK), and the File Restoration Utility (RMSRST) has been modified to append summary listings of processing to a single file if that file exists.
- Five switches have been added to the RMS-11 File Conversion Utility (RMSCNV): the /CA switch, the /EO switch, the /IM switch, the /KN switch, and the /ML switch. The /CA switch creates an output file with the attributes of an existing file; the /EO switch changes ASCII stream files that end

with a CTRL/Z character to end with null characters; the /IM switch reads and writes records in block mode; the /KN switch reads an indexed file using a key name; and the /ML switch explicitly sets the limit of memory allocation. The /PD switch no longer requires a number sign (#) before an octal padding character or quotation marks around an ASCII padding character unless it is a lowercase character, a space, or a tab.

- In addition, RMSCNV file conversion operations can now specify files on remote nodes. On systems with DECnet network capabilities and on which RMSCNV has Data Access Protocol (DAP) support, RMSCNV file conversions can take place not only locally, but over the network as well. You can convert files between a local and a remote node, or between two remote nodes.

- One switch has been added to the RMS-11 File Restoration Utility (RMSRST): /CV. The /CV switch converts file version numbers from octal to decimal for files created and backed up on RSX-11M/M-PLUS systems and restored to VAX/VMS systems; and from decimal to octal for files created and backed up on VAX/VMS systems and restored to RSX-11M/M-PLUS systems.

- All utilities now display information in lowercase letters, and the error message format for all utilities (except RMSDEF) has been changed.

2.3 COMPARISONS WITH RMS-11 VERSION 1.8

Conversion from RMS-11 V1.8 to RMS-11 V2.0 has been made transparent where possible. Structural changes required by feature additions, however, may be significant to some programs.

2.3.1 Memory-Resident Library (RMSRES)

RMSRES now contains all RMS-11 support code save for minor linkage code in the task. It has been configured as multiple segments, not all of which need be resident in memory simultaneously, thus dramatically reducing instantaneous physical memory overhead. The library is also:

- "Task-independent," which means that it can be rebuilt or replaced and the tasks that use it need not be rebuilt

- "Clusterable," which means that it can share task virtual memory with other libraries that support "clustering" (see your FMS and language documentation)

These features required changes in the RMSRLX.ODL file, the file normally used to build tasks using RMSRES. Tasks that used RMSRLX.ODL with RMS-11 V1.8 should continue to build properly using the new RMSRLX.ODL and RMSRES. Build files that used RMSRES but did not use RMSRLX.ODL with RMS-11 V1.8 must be changed to use (or incorporate the contents of) the new RMSRLX.ODL file.

2.3.2 Memory-Resident Library (RMSSEQ)

RMS-11 V2.0 contains no library equivalent to the RMSSEQ memory-resident library included with RMS-11 V1.8. RMSSEQ offered no advantages in use of task virtual memory, and no advantages in use of system physical memory which are not equally attainable through use of disk-overlaid RMS-11 at negligible performance cost. In addition, structural constraints made it impossible for RMSSEQ to be clusterable or task-independent.

The improved packaging of the full-function library RMSRES should make it a suitable replacement for RMSSEQ in most environments. Where this is inappropriate, disk-overlaid or non-overlaid RMS-11 can be used.

2.3.3 Memory-Resident Library (DAPRES)

If your system includes the necessary DECnet facilities, the new resident library DAPRES provides support for remote file and record access as well as local access. Like RMSRES, DAPRES is both task-independent and clusterable. In particular, a task using RMSRES may include remote-access support via DAPRES with negligible increase in task virtual address space requirement by clustering the two libraries: the file DAPRLX.ODL should be used to generate this configuration.

2.3.4 Disk-Overlaid RMS-11

Structural changes to the RMS-11 code were performed so that the RMS-11 V1.8 ODL files RMS11X, RMS12X, and RMS11S would remain valid. Most other ODL structures will probably continue to be valid, although validity cannot be guaranteed for all cases. Use of V1.8 ODL structures with V2.0 code may require more task virtual address space than with V1.8 code.

The new versions of RMS11X, RMS12X, and RMS11S use approximately the same (or slightly less) task virtual address space as their V1.8 counterparts used with the V1.8 code. If your system includes the necessary DECnet facilities, the new file DAP11X.ODL provides support for remote file and record access as well as local access.

The ODL MAKRMSODL.CMD is not provided and should not be used with V2.0 of RMS-11. However, a prototype ODL file, RMS11.ODL, is provided, and contains extensive comments. This file is not meant to be used directly, but may be helpful in designing your own ODL structures or in modifying the structure of the above ODL files to suit your special needs.

A restriction on ODL structures involves the RMS-11 entry point modules R0xxxx (where **xxxx** represents the leading characters of the RMS-11 operation name, for example, R0PUT). These modules, which were not explicitly referred to in previous RMS-11 ODL files, must **not** be referred to in RMS-11 ODL files for V2.0; if they are, a multiple definition for symbols of the form $RMxxx will occur.

2.3.5 Non-Overlaid RMS-11

While the total end-to-end size of RMS-11 has decreased despite addition of new features, the space savings and feature additions have not been spread evenly throughout. Most of the added features have occurred in the areas of file/directory operations and record operations for sequential files.

The result is that applications using non-overlaid RMS-11 for sequential-file-only access may experience an increase in virtual address space requirements for the RMS-11 code. If this presents a problem, it should normally be possible to select a suitable RMS-11 disk-overlay structure without noticeable performance impact.

2.3.6 Other RMS-11 Version 1.8 Compatibility Notes

- The utility RMSDFN is no longer supported.

- There is no longer a distinction between "standard" RMS-11 and "11K" RMS-11. All RSX-11M/M-PLUS systems automatically include the appropriate files needed for RMS-11, including indexed support.

- Programs that have been compiled under RMS-11 V1.8 should in virtually all cases still run under V2.0, without recompilation when relinked. Such programs, however, will not be able to take advantage of any new features in Version 2.0 of RMS-11.

- Tasks already built using RMS-11 V1.8 should continue to run correctly (without rebuilding) on RSX-11M-PLUS V2.1. However, it is possible that RMS-11 V1.8 (or prior) tasks will not execute correctly on versions of RSX-11M-PLUS **after** V2.1. RMS-11 V2.0 has been implemented to anticipate this possible change in RSX-11M-PLUS: this allows you one full release cycle to convert any such tasks to RMS-11 V2.0.

- In general, RMS-11 V2.0 cannot be run on a previous version of RSX-11M-PLUS.

  RMS-11 V2.0 depends on new features included in the F11ACP and on the new structure of TKB autoload vectors in V2.1 of RSX-11M-PLUS; do not try to use old versions of TKB to link RMS-11 V2.0 programs.

  Note: The reverse situation can also cause certain problems. Because of the change in TKB autoload vector format, a task that was linked with new TKB but old RMS-11 (V1.8 or before), and that uses RMS-11 asynchronously, will behave unpredictably.

- With RMS-11 V2.0, some changes to the SYSLIB modules have been made; several modules have been placed in SYSLIB to maintain compatibility with previous releases.

  As a result, you must be careful to reference RMSLIB as a "search" library before referring to SYSLIB when building tasks that use non-overlaid RMS-11. (This is only necessary if you need to make an explicit reference to SYSLIB as a "search" library when task-building.)

- MACRO-11 programs using RMS-11 should always check for errors by checking the value of the appropriate STS field.

  With V1.8, the C-bit may have been returned on RMS-11 errors. However, this is not a defined error return and has not been maintained in RMS-11 V2.0.

- <CTRL/Z> and <ESC> are no longer recognized as record terminators for stream files, and <CTRL/Z> is no longer recognized as a file terminator for stream files.

  Note that RMSCNV has a new switch, /EO, which can be used to convert stream files which were created by a non-RMS-11 program, and which used <CTRL/Z> as the file terminator.

- The expanded string is now an output from the OPEN and ERASE operations by file-id.

- RMS-11 now makes a check that at least 2 keys fit into a bucket. Therefore, with a bucket size of 1, the maximum key size is 243.

- When block I/O is used, OPEN and DISPLAY operations no longer return relative or indexed file key XAB, allocation XAB, or summary XAB information.

2.4 MISCELLANEOUS INFORMATION

- Note that all the new features provided in RMS-11 are fully accessible only to the MACRO-11 programmer; for high-level languages, check the language documentation for supported features.

- For magnetic tape, RMS-11 now allows ANSI fixed format tape records to be less than 18 bytes. In addition, RMS-11 supports the full set of ANSI tape names as quoted strings. This includes the full set of ANSI 'a' characters, up to 17 bytes.

- Several RMS-11 demonstration programs have been included on the distribution kit. These sample programs are provided as MACRO-11 sources and task images; they illustrate how to use the new PARSE/SEARCH/RENAME/ERASE operations and wildcarding facilities. A GSA source routine is also included to illustrate how to extend the task in the event of RMS-11 pool exhaustion.

  The names of the programs are PARSE, SEARCH, RENAME, ERASE, and GSA.

- The RMSIFL and RMSDES utilities and the remote access (DAP) support code in RMS-11 V2.0 will run only on CPUs that support the EIS instruction set.

2.5 PROBLEMS FIXED WITH RMS-11 VERSION 2.0

- The problem that was reported for RMS-11 V1.8 on GET after FIND on alternate keys (Dispatch article number 48.2.20) has been fixed for RMS-11 V2.0.

- RMSIFL V1.8 did not load indexed files correctly if the key type was integer. (Not all records could be accessed by key value.) This problem has been fixed in the new version of RMSIFL.

- RMS-11 now pads stream files with nulls, to the high block of the file (not just to the end of the current block). This will make RMS-11-created stream files readable by programs that do not recognize the EOF value from the file header.

2.6 KNOWN PROBLEMS WITH RMS-11 VERSION 2.0

- Under some circumstances, the deferred write option may be ignored.

- Use of the mass-insertion feature may occasionally cause the error ER$DME.

- When you use RMSDES and specify a GET command, RMSDES incorrectly requests write access on the GET file. In some cases, an ER$FLK or ER$PRV error may result if your task does not have write access to the file.

2.7 RMS-11 FILES AND PLACEMENT ON THE DISTRIBUTION KIT

The following table describes the contents of the RMS-11 V2.0 distribution kit.

Note that some file names are marked with an asterisk. This indicates that the files are not included on RL02 and RC25 distribution kits. They are excluded because of space reasons on the kit, and because they are only used to rebuild components of RMS-11 V2.0.

File Name	UIC	Comments
RMSMAC.MLB	LB:[1,1]	Can be deleted if you are not using MACRO-11 RMS-11 programs.
RMSLIB.OLB	LB:[1,1]	Object library for RMS-11 local access.
RMSDAP.OLB	LB:[1,1]	Object library for RMS-11 remote access. Can be deleted if you do not need remote access.
RMSBCK.TSK	LB:[3,54]	RMSBCK utility; uses RMSRES.
RMSRST.TSK	LB:[3,54]	RMSRST utility; uses RMSRES.
RMSDEF.TSK	LB:[3,54]	RMSDEF utility; uses RMSRES.
RMSDSP.TSK	LB:[3,54]	RMSDSP utility; uses RMSRES.
RMSCNV.TSK	LB:[3,54]	RMSCNV utility; uses RMSRES.
RMSDES.TSK	LB:[3,54]	RMSDES utility; uses RMSRES.
RMSIFL.TSK	LB:[3,54]	RMSIFL utility; uses RMSRES.

(Continued on next page)

File Name	UIC	Comments
RMSDES.IDX	LB:[1,2]	Indexed help file used by RMSDES.
RMS11.ODL	LB:[1,1]	Prototype ODL file.
R0RMS1.MAC	LB:[1,1]	For use with the prototype ODL.
RMS11S.ODL	LB:[1,1]	ODL file for sequential.
RMS12S.ODL	LB:[1,1]	ODL file for sequential.
RMS11X.ODL	LB:[1,1]	Standard indexed file ODL.
RMS12X.ODL	LB:[1,1]	Indexed file ODL.
RMSRLX.ODL	LB:[1,1]	ODL for use with RMSRES.
DAP11X.ODL	LB:[1,1]	ODL for use with overlaid RMSDAP.
DAPRLX.ODL	LB:[1,1]	ODL for use with DAPRES.
RMSRES.TSK	LB:[1,1]	TSK image for linking against RMSRES, as a non-supervisor-mode library.
RMSLBA.TSK	LB:[1,1]	Task image for RMSRES segment.
RMSLBB.TSK	LB:[1,1]	Task image for RMSRES segment.
RMSLBC.TSK	LB:[1,1]	Task image for RMSRES segment.
RMSLBD.TSK	LB:[1,1]	Task image for RMSRES segment.
RMSLBE.TSK	LB:[1,1]	Task image for RMSRES segment.
RMSLBF.TSK	LB:[1,1]	Task image for RMSRES segment.
RMSRES.STB	LB:[1,1]	STB file for RMSRES.
DAPRES.TSK	LB:[1,1]	Task image for RMSDAP resident library.
DAPRES.STB	LB:[1,1]	STB file for DAPRES.
RMSLBA.MAP*	LB:[1,34]	Maps for segmented library.
RMSLBB.MAP*	LB:[1,34]	
RMSLBC.MAP*	LB:[1,34]	
RMSLBD.MAP*	LB:[1,34]	
RMSLBE.MAP*	LB:[1,34]	
RMSLBF.MAP*	LB:[1,34]	
DAPRES.MAP*	LB:[1,34]	
RMSRES.TSK	LB:[3,54]	Task image for linking in supervisor mode. Also installed as the root of the library (supervisor mode and non-supervisor mode).
RMSRES.STB	LB:[3,54]	STB file for linking supervisor-mode tasks.
RMSRES.MAP*	LB:[3,54]	Map file for supervisor-mode library.
RMSFAK.CMD*	LB:[1,24]	For rebuilding RMSRES.
RMSROT.STB*		
RMSROT.CMD*		
RMSZAP.CMD*		
RMSLBA.CMD*	LB:[1,24]	For rebuilding RMSLBA.
RMSLBA.ODL*		
RMSLBB.CMD*	LB:[1,24]	For rebuilding RMSLBB.
RMSLBB.ODL*		
RMSLBC.CMD*	LB:[1,24]	For rebuilding RMSLBC.
RMSLBC.ODL*		
RMSLBD.CMD*	LB:[1,24]	For rebuilding RMSLBD.
RMSLBD.ODL*		
RMSLBE.CMD*	LB:[1,24]	For rebuilding RMSLBE.
RMSLBE.ODL*		
RMSLBF.CMD*	LB:[1,24]	For rebuilding RMSLBF.
RMSLBF.ODL*		

(Continued on next page)

File Name	UIC	Comments
DAPRES.CMD* DAPRES.ODL*	LB:[1,24]	For rebuilding DAPRES.
BCKNON.CMD* BCKNON.ODL*	LB:[1,24]	For rebuilding the overlaid version of RMSBCK.
BCKNRN.CMD* BCKNRN.ODL*	LB:[1,24]	For rebuilding the resident-library version of RMSBCK.
BCKNSN.CMD* BCKNSN.ODL*	LB:[1,24]	For rebuilding the supervisor-mode version of RMSBCK.
CNVNON.CMD* CNVNON.ODL*	LB:[1,24]	For rebuilding the overlaid version of RMSCNV.
CNVNRN.CMD* CNVNRN.ODL*	LB:[1,24]	For rebuilding the resident library version of RMSCNV.
CNVNOO.CMD* CNVNOO.ODL*	LB:[1,24]	For rebuilding the overlaid version of RMSCNV, with RMSDAP.
CNVNRR.CMD* CNVNRR.ODL*	LB:[1,24]	For rebuilding the resident library version of RMSCNV,with RMSDAP.
CNVNSN.CMD* CNVNSN.ODL*	LB:[1,24]	For rebuilding the supervisor-mode version of RMSCNV.
DEFNON.CMD* DEFNON.ODL*	LB:[1,24]	For rebuilding the overlaid version of RMSDEF.
DEFNRN.CMD* DEFNRN.ODL*	LB:[1,24]	For rebuilding the resident library version of RMSDEF.
DEFNSN.CMD* DEFNSN.ODL*	LB:[1,24] LB:[1,24]	For rebuilding the supervisor mode version of RMSDEF.
DESNON.CMD* DESNON.ODL*	LB:[1,24]	For rebuilding the overlaid version of RMSDES.
DESNRN.CMD* DESNRN.ODL* RMSDES.ODL*	LB:[1,24]	rebuilding the resident library version of RMSDES.
DESNSN.CMD* DESNSN.ODL*	LB:[1,24]	For rebuilding the supervisor-mode version of RMSDES.
DSPNON.CMD* DSPNON.ODL*	LB:[1,24]	For rebuilding the overlaid version of RMSDSP.
DSPNRN.CMD* DSPNRN.ODL*	LB:[1,24]	For rebuilding the resident-library version of RMSDSP.
DSPNSN.CMD* DSPNSN.ODL*	LB:[1,24]	For rebuilding the supervisor-mode version of RMSDSP.
IFLNON.CMD* IFLNON.ODL*	LB:[1,24]	For rebuilding the overlaid version of RMSIFL.
IFLNRN.CMD* IFLNRN.ODL*	LB:[1,24]	For rebuilding the resident-library version of RMSIFL.
IFLNSN.CMD* IFLNSN.ODL*	LB:[1,24]	For rebuilding the supervisor-mode version of RMSIFL.
RSTNON.CMD* RSTNON.ODL*	LB:[1,24]	For rebuilding the overlaid version of RMSRST.
RSTNRN.CMD* RSTNRN.ODL*	LB:[1,24]	For rebuilding the resident-library version of RMSRST.
RSTNSN.CMD* RSTNSN.ODL*	LB:[1,24]	For rebuilding the supervisor-mode version of RMSRST.
RMSUTL.OLB* RMSODL.ODL*	LB:[1,24]	For rebuilding several utilities

(Continued on next page)

File Name	UIC	Comments
GSA.MAC	LB:[200,1]	Demonstration program included as an illustration of how to extend an RMS-11 task in the event of pool exhaustion.
PARSE.MAC, SEARCH.MAC, RENAME.MAC, ERASE.MAC, PARSE.TSK, SEARCH.TSK, RENAME.TSK, ERASE.TSK	LB:[200,1]	Demonstration programs for the new directory and wildcarding facilities

Note that all RMSDAP files can be deleted if you are not using RMS-11 to access files on remote nodes.

2.8 RMS-11 VERSION 2.0 INSTALLATION

On RSX-11M-PLUS systems, all RMS-11 files are automatically on your system after system generation.

2.8.1 Startup Command Procedures

To install RMS-11, all you need do is install the resident libraries and RMS-11 utilities at system startup. To aid you in installing them, the file LB:[1,2]STARTUP.CMD contains commands and sample comments that can be edited to become system startup commands. Some items to note:

- On RL02 and RC25 pregenned systems, the RMS-11 segmented library (RMSRES, RMSLBA through RMSLBF), and all the RMS-11 utilities are already installed in the system image. For these kits, the only installation that is needed is the optional installation of the DAPRES resident library, if the system has DECnet support and RMS-11 remote access facilities are to be used.

- On systems other than RL02 and RC25, the startup file contains commands that install the segmented resident library in the system image. Note that these are commands, not comments; if you do not want to install the library, you should edit the file to make these commands comments. Some items to note:

  - All resident libraries should be installed using the option /RON=YES.

  - The root of the library, RMSRES, must be the task image contained in [3,54]. The [1,1]RMSRES.TSK should never be installed in the system; it is only used when tasks are linked against the nonsupervisor mode version of RMSRES.

  - If the root of the segmented library, RMSRES, is installed in the system, you **must** install the remaining library tasks (RMSLBA through RMSLBF).

Failure to install the root segment RMSRES will cause the error "INS - Common block not Loaded RMSRES" when a referencing task or utility is invoked; however, failure to install any of the remaining resident library segments will not give an error at invocation, but may cause a BPT trap (with R0 containing ER$LIB) when the missing segment is subsequently needed by RMS-11.

NOTE

There are several circumstances involving "inconsistent or incomplete resident libraries" which can cause a BPT trap to be generated, with R0 containing the error code ER$LIB. This can happen if not all the segments of the library are installed or if the version numbers of one or more segments do not match the root segment, the RMSDAP code, or the task itself. In particular this can happen to the RMS-11 utilities if they are built against the segmented resident library and the resident library segments are installed incorrectly.

- The resident libraries have been built with the PAR=parname option in the TKB command file, where **parname** is the name of the resident library. (This feature was included for compatibility with RSX-11M systems.)

  However, you should install the resident libraries in the GEN partition. Do **not** generate individual partitions for the libraries on RSX-11M-PLUS; doing so would negate much of the benefit of having a "demand-paged" segmented library.

- If you are using RMSRES in supervisor mode, you do **not** need to install a different task. You simply need to link your task as described in Section 2.1.3, Supervisor-Mode Support.

- If you want to use the RMS-11 utilities or the RMSDAP resident library (RMSDAP), you must edit the startup command file, which contains template commands (comments) that you can edit to install these facilities. Note that if you want to use the RMS-11 utilities, you must have installed the complete segmented resident library (as described above).

2.8.2 Utility Configurations

The RMS-11 utilities that are provided on the distribution kit are built to use the segmented resident library. Consequently, RMSRES, RMSLBA, RMSLBB, RMSLBC, RMSLBD, RMSLBE, and RMSLBF must all be installed before you use any of the RMS-11 utilities.

2.8.2.1 **Utility Command and ODL Files** - For each utility, a command and ODL file is provided which can be used to build the utility using disk-overlaid RMS-11. The names of the files are the following:

```
For RMSBCK:  BCKNON.CMD,BCKNON.ODL,RMSODL.ODL
For RMSRST:  RSTNON.CMD,RSTNON.ODL,RMSODL.ODL
For RMSCNV:  CNVNON.CMD,CNVNON.ODL
For RMSDSP:  DSPNON.CMD,DSPNON.ODL,RMSODL.ODL
For RMSDES:  DESNON.CMD,DESNON.ODL,RMSDES.ODL
For RMSDEF:  DEFNON.CMD,DEFNON.ODL,RMSODL.ODL
For RMSIFL:  IFLNON.CMD,IFLNON.ODL
```

NOTE

These command and ODL files are not supplied with the RL02 and RC25 kits.

For RMSCNV, two additional configurations are available, if you want to use RMSCNV to access files on remote nodes. To build RMSCNV using the clustered RMS-11 and RMSDAP resident libraries, you can use

```
For RMSCNV:  CNVNRR.CMD,CNVNRR.ODL
```

To build RMSCNV using disk-overlaid RMS-11 and RMSDAP, you can use

```
For RMSCNV:  CNVNOO.CMD,CNVNOO.ODL
```

For each utility, a command and ODL file are provided which can be used to build the utility using the resident library RMSRES. The names of the files are the following:

```
For RMSBCK:  BCKNRN.CMD,BCKNRN.ODL
For RMSRST:  RSTNRN.CMD,RSTNRN.ODL
For RMSCNV:  CNVNRN.CMD,CNVNRN.ODL
For RMSDSP:  DSPNRN.CMD,DSPNRN.ODL
For RMSDES:  DESNRN.CMD,DESNRN.ODL
For RMSDEF:  DEFNRN.CMD,DEFNRN.ODL
For RMSIFL:  IFLNRN.CMD,IFLNRN.ODL
```

For each utility, a command and ODL file are provided which can be used to build the utility using the supervisor-mode library RMSRES. The names of the files are the following:

```
For RMSBCK:  BCKNSN.CMD,BCKNSN.ODL
For RMSRST:  RSTNSN.CMD,RSTNSN.ODL
For RMSCNV:  CNVNSN.CMD,CNVNSN.ODL
For RMSDSP:  DSPNSN.CMD,DSPNSN.ODL
For RMSDES:  DESNSN.CMD,DESNSN.ODL
For RMSDEF:  DEFNSN.CMD,DEFNSN.ODL
For RMSIFL:  IFLNSN.CMD,IFLNSN.ODL
```

2.8.2.2 **Rebuilding the Utilities** - To rebuild the utilities, you must take the following steps:

- Log in to a privileged account.
- Set your default account to [1,24] on the system disk.
- Use TKB to build the utility or utilities.

The utilities and the corresponding map files will be built in the current account; you may then want to put the utility in a system account ([3,54]).

Each command file for a particular utility creates a utility of the correct name. So, for example, CNVNRN.CMD and CNVNON.CMD both produce task images called RMSCNV.TSK; one is built against the resident library and one is built with disk-overlaid RMS-11.

Finally, note that the RMSIFL utility command files include an EXTTSK value that is used to increase the size of internal buffers, thereby allowing RMSIFL to load larger-sized files and files with many area and key definitions.

The default EXTTSK which is specified is fairly large. If you want a smaller version of RMSIFL, you can simply install RMSIFL with a smaller increment using the /INC switch. The /INC switch will override the EXTTSK directive.

The EXTTSK value however, must be greater than 2048 (decimal) words. If the value is too small, you may get the error "IFL memory exhausted," or a SORTs error code.

2.9 REPORTING PROBLEMS

Software Performance Reports (SPRs) allow you to report any problems directly to DIGITAL. The general procedure for submitting an SPR is described in Sections 1.4 and 1.4.1. The following additional information should be submitted with SPRs on RMS-11:

1. Include the version number and patch level of the RMS-11 that you are using.

2. Indicate whether you are using a programming language to process the file(s) and include the version number and patch level of the language.

3. If RMS-11 aborts (or if some other task crashes and RMS-11 appears to be the cause), include a post-mortem dump and a map of the task involved.

4. Include copy(s) of the file(s) involved, in RMSBCK format.

5. If the errors are reproducible, include copies of the files that can cause the error. If the problem is not reproducible, include a copy of the corrupt file if possible.

6. For RMS-11 utilities, include also a description of the command line(s) or interactive session that led to the error.

   Include a listing of the actual error that occurred, if possible.

READER'S COMMENTS

NOTE: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well organized? Please make suggestions for improvement.

Did you find errors in this manual? If so, specify the error and the page number.

Please indicate the type of user/reader that you most nearly represent.

- ☐ Assembly language programmer
- ☐ Higher-level language programmer
- ☐ Occasional programmer (experienced)
- ☐ User with little programming experience
- ☐ Student programmer
- ☐ Other (please specify) _____

Name _____ Date _____

Organization _____

Street _____

City _____ State _____ Zip Code _____
or Country

— — Do Not Tear - Fold Here and Tape — — — — — — — — — — — — — — — — — —

digital

No Postage
Necessary
if Mailed in the
United States

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

BSSG PUBLICATIONS ZK1-3/J35
DIGITAL EQUIPMENT CORPORATION
110 SPIT BROOK ROAD
NASHUA, NEW HAMPSHIRE 03061

— — Do Not Tear - Fold Here — — — — — — — — — — — — — — — — — —

Cut Along Dotted Line